Volume 9 • Issue 5 • September / October 2015

Editor-in-Chief
Dr. Chen-Chi Shing

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND SECURITY (IJCSS)

ISSN : 1985-1553

Publication Frequency: 6 Issues / Year

CSC PUBLISHERS
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND SECURITY (IJCSS)

VOLUME 9, ISSUE 5, 2015

EDITED BY
DR. NABEEL TAHIR

ISSN (Online): 1985-1553
International Journal of Computer Science and Security is published both in traditional paper form and in Internet. This journal is published at the website http://www.cscjournals.org, maintained by Computer Science Journals (CSC Journals), Malaysia.

IJCSS Journal is a part of CSC Publishers
Computer Science Journals
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND SECURITY (IJCSS)

Book: Volume 9, Issue 5, September / October 2015
Publishing Date: 31-10-2015
ISSN (Online): 1985 -1553

IJCSS Journal is a part of CSC Publishers
http://www.cscjournals.org

© IJCSS Journal
Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers, 2015

EDITORIAL PREFACE

This is *Fifth* Issue of Volume *Nine* of the International Journal of Computer Science and Security (IJCSS). IJCSS is an International refereed journal for publication of current research in computer science and computer security technologies. IJCSS publishes research papers dealing primarily with the technological aspects of computer science in general and computer security in particular. Publications of IJCSS are beneficial for researchers, academics, scholars, advanced students, practitioners, and those seeking an update on current experience, state of the art research theories and future prospects in relation to computer science in general but specific to computer security studies. Some important topics cover by IJCSS are databases, electronic commerce, multimedia, bioinformatics, signal processing, image processing, access control, computer security, cryptography, communications and data security, etc.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Started with Volume 9, 2015, IJCSS appears with more focused issues. Besides normal publications, IJCSS intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

This journal publishes new dissertations and state of the art research to target its readership that not only includes researchers, industrialists and scientist but also advanced students and practitioners. The aim of IJCSS is to publish research which is not only technically proficient, but contains innovation or information for our international readers. In order to position IJCSS as one of the top International journal in computer science and security, a group of highly valuable and senior International scholars are serving its Editorial Board who ensures that each issue must publish qualitative research articles from International research communities relevant to Computer science and security fields.

IJCSS editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCSS. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. IJCSS provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts.

Editorial Board Members
International Journal of Computer Science and Security (IJCSS)

TABLE OF CONTENTS

Volume 9, Issue 5, September / October 2015

Pages

A Mitigation Technique For Internet Security Threat of Toolkits Attack

Francisca Nonyelum Ogwueleka *ogwuelekefn@fuwukari.edu.ng*
Computer Science Department,
Federal University Wukari,
PMB 1020 Wukari,
Taraba State, Nigeria.

Edward Onyebueke Agu *aguedward@fuwukari.edu.ng*
Computer Science Department,
Federal University Wukari,
PMB 1020 Wukari,
Taraba State, Nigeria.

Abstract

The development of attack toolkits conforms that cybercrime is driven primarily by financial motivations as noted from the significant profits made by both the developers and buyers. In this paper, an enhanced hybrid attack toolkit mitigation model was designed to tackle the economy of the attack toolkits using different techniques to discredit it. The mitigation looked into Zeus, a common and the most frequently used attack toolkit to discover the hidden information used by the attackers to launch attacks. This information helped in creating honey toolkits, honeybot and honeytokens. Honeybots are used to submit honeytoken to botmasters, who sells to the internet black market. Both the botmasters, his mules and buyers attempts to steal huge amount of money using the stolen credentials which includes both real and honeytokens and will be detected by an attack detector which sends an alert on any transaction involving the honeytokens. A reconfirmation process which is secured using enhanced RC6 cryptosystem is enacted. The reconfirmation message in plain text is securely encrypted into cipher text and transmitted from the bank to the legitimate account owner and vise visa. The result of the crypto analysis carried out on the encrypted text using RC6 encryption algorithm showed that the cipher text is not transparent.

Keywords: Toolkits, Mitigation, Zeus, Botmaster, Cryptosytem, Botnets.

1. INTRODUCTION

Internet has been considered to be universal in scope, but its open medium is not secure. Internet has transformed the computing and communications world in order to develop and support client and server services. The existence of internet browsers and internet technologies has increased the efficiency and effectiveness of the internet service to the users throughout the whole world [1]. Today, The Internet consists of infrastructures and mechanisms for sharing information across the globe [1].

Due to many activities being carried out on the internet ranging from electronic mail to sharing of professional knowledge, has led to internet insecurity problems which is of high importance. Internet problems need critical attention, unsecured online businesses on which many lives depend for survival draws the attention of online criminals [1].

A packs of mischievous codes used to carryout rigorous and extensive attack on computers in a network is called attack toolkits or crimeware [2]. Attack toolkits consist of established mischievous codes for taking advantages of weakness in devices together with many other tools to modify, install, and computerize extensive attacks, which includes command-and-control (C&C)

attack at the server [3]. Attack toolkits are used for stealing vital information or to change attacked systems to zombie bots or network (botnet) so as to achieve more attacks according to [4]. The advertisement and selling of these attack toolkits are on the internet secretive black market which consist of illegal transactions among internet criminals. The ease-of-use nature of these toolkits has played a major role in turning internet crime to a lucrative means of acquiring wealth [2].

Some toolkits attack codes focuses on specific program while others have sophistical codes to carryout multiple attack on many programs across many operating platforms [2]. The combined effort of these attack kits and Internet-network controlled from a remote location is to performed mischievous and illegal acts on life supporting information and equipment. [5].

The mitigation techniques in this research showcase a number of ways that can be used to curb this insecurity irrespective of the cadre. Some part of this mitigation approach inter-worked together and appeared as aspect of the general approach for toolkits attack mitigation. [6].

2. REVIEW OF RELATED STUDIES

Determining botnets membership using Domain Name System (DNS) blacklist counter intelligence was proposed by [7], but it is only effective to certain categories of spam botnets. A system to sense a botnet using network traffic aggregation called Tand system, was proposed by [8]. This is not too effective to detect botnets as botnets changes their mode of attack frequently to circumvent this solution. Similarly, a system for passive detection of individual attack toolkits using correlation of Intrusion Detection System (IDS) alerts to a predefined infection model called BotHunter was proposed by [9]. [10] proposed a system which correlates network traffic to detect botnets called BotMiner, but with different parameters to Tand system. A system to detect Internet Relay Chat (IRC) botnets using signature detection on known IRC nicknames called Rishi was proposed by [11]. All these measures face the same limitation as Tand system, because botnets constantly changes their mode of operation in order to escape these measures.

So many researchers have equally done a great research works on the field of identity theft analysis, detection and mitigation. Two attacks on the trust and rating system of the internet black market was presented by [12], but their method has limitation as it did not handle the case where credentials are not sold on the black markets. Research on detection of attack sites by submitting fake credentials and monitoring site behaviour was presented by [6], but their approach did not mitigate attack toolkit. A research to analyzed data that is sized from 70 drop zones in order to provide metrics to determine the wealth of underground economy was presented by [13]. A system that used honeytokens to track attackers was presented by [14]. All these measures did not submit the honeytokens through attack toolkit and did not mitigate them. A system to destabilized Distributed Denial of Service (DDoS) attack using honeypots was created by [15]. The attack was to scatter the profit margins of botmasters by reducing the value of their services. While all these measures show good steps to detect and mitigate internet attack, they have not addressed the malware authors or the botmasters that created the attack toolkits. The enhance hybrid model in this research moved further to extend the economic attack to lower the botmaster’s profits by aiming the kits they are selling. The closest work to this model is a framework proposed by [16] and [17]. [16] in approach utilize spam traps to send credentials to phishing sites as well as using phoneybots to send credentials to botnets such as Zeus. But the main aim is the end-users or the money mules, while [17] in their approach targeted to disrepute a specific kit and the creator of the toolkit. The model in this research will be a hybrid of the two frameworks by complementing the limitations of each one with the strength of another. We will adopt more security features such as encryption to further improve the hybrid model. This hybrid method stands to bring to minimum the variation of these kits that is accessible to cybercriminal and directly affect their extensive uses.

2.1 The First Framework by [17]

The ultimate aim of [17] in their proposed framework is to damage the reputation of a botnet toolkit and reduce its demand. They attack integrity of the toolkit in two ways: the toolkit profitability with respect to its use to sale credentials and the security of the toolkit users from prosecution. This research proposed two stages to this approach: to reduce a toolkit's profitability by submitting false credentials to it, and to make the toolkit insecure by submitting honeytokens to the botnets which will help in arresting and prosecuting the end-users of the stolen credentials. The framework which shows the two stages and its process is illustrated in Figure 1.

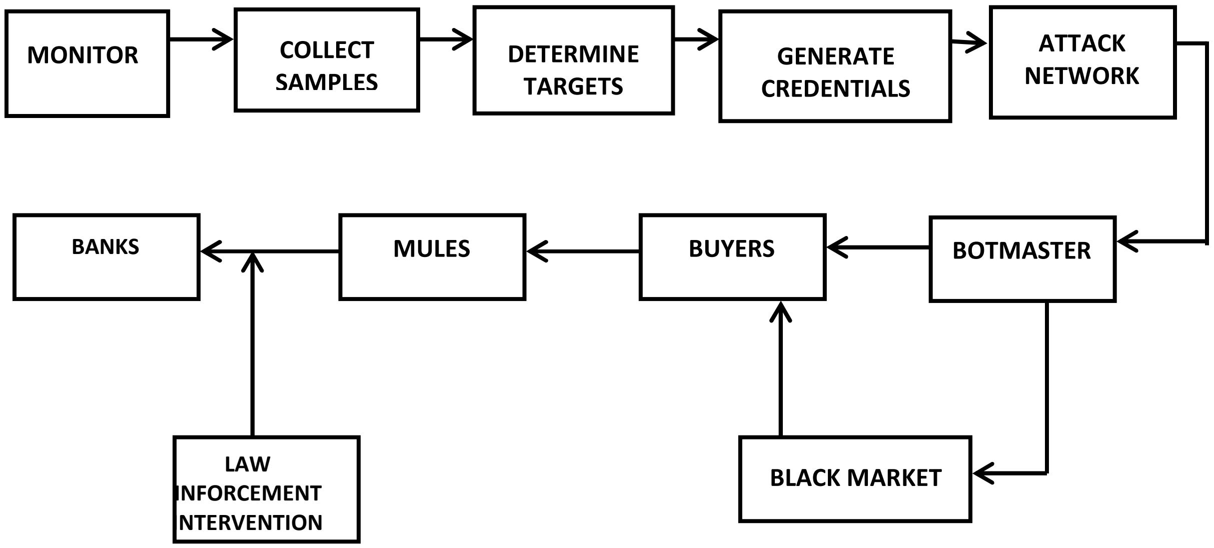

FIGURE 1: Botnet toolkit attack and detection process [17].

The first stage aimed at mixing the stolen credentials from a botnet with honey tokens have the following steps -

i. It monitors the current activities of the botnet toolkit and selects the most used toolkit for stealing credentials as their target.
ii. It collects many samples of botnet binaries from honeynets, antivirus companies, financial institutions and security forums.
iii. It analyzes the toolkit samples to determine the targeted websites and their targeted fields.
iv. It creates false credentials.
v. It submits the credentials to the botmasters using the toolkit.

The first stage of the frameworks ends at this point, and the false credentials are sent to the internet black market.

In the second stage of the framework which is an expansion of the first framework, aimed at discrediting the security of the end-users of the stolen credentials. The stage contains one additional step which takes place after the credentials have been sold and passes via the internet black market: The second stage of the framework monitors honey account usage in order to make arrests when money are withdrawn from the account and purchases made using the honey credentials are received. This framework only targets to damage the reputation of a toolkit and the authors that made them.

2.2 The Second Framework by [16]

The anti-phishing framework proposed is an organized solution showcasing steps used in the framework information flow. The steps are -

Step 1: The honeyed bank for example, sets up a number of virtual machines (m >= 1), running 'm' honeybots and 'm' spam traps. Each spam trap or honeybot is fed with a honeytoken. Following the normal characteristics of all bank customers, each honeybot accesses e-banking system frequently and carryout virtual online transactions. A designated group of personnel in a bank take the responsibility of managing the virtual machines, spamtraps and honeybots. They also monitor doubtful phishing e-mails.

Step 2: The e-criminal sends threat e-mails to his prospective victims, or contaminate prospective victims' computers with attack bot, or launches a sophisticated attacks.

Steps 3: During phishing attack using threat e-mails, the group of managers in the honey bank submits the tokens accompanied with the honeytokens to the attack site. In a sophisticated phishing attack based on attack toolkit, the e-criminal collects automatically both tokens and honeytokens each time attack toolkit is used. The collected tokens are used to access the e-banking system. Each honeytoken should be submitted only once after which a new honeytoken will be created and assigned to attack toolkit by the group of managers after the old honeytoken is used up. Each toolkit should be updated after submitting a honeytoken to avoid it being detected.

Step 4: The e-criminal gets some valid identities that is mixed with some honeytokens from the attack website, or from the attack toolkit.

Step 5: The e-criminal visits the honeyed e-banking system with the credentials or honeytokens that have been collect. They may wish to examine the validity of the honeytokens, and it is more reasonable, to assume that the botmaster knows how the framework works.

Step 6: The botmaster launches attack by trying to steal money from both the victims' accounts and the honeytokens.

Step 7: A detector placed in the honeyed e-banking system checks all e-transactions with respect to all false credentials or honeytokens. It sends a signal immediately it detects an attack by botmaster trying to steal a good or predefined amount of money from a honeytoken to a non-honeytoken account. The detector marks the receiver's account to be highly doubtful or suspicious which is in turn regarded as "phishing account". This phishing account is usually a mule's account opened and operated in another bank.

Step 8: The affected bank requests for the consent of the prospective victims or the prospective victims contact the bank, for reconfirmation of all money that are transferred from victims accounts to any other account that have being marked to be phishing accounts.

Step 9: The prospective victims then have the right to agree or refuse the transactions through the reconfirmation message. For the case of rejection, their account may be locked temporarily, and their credentials may be reset.

Steps 10: The bank informs appropriate authorities about the doubtful attack activities, it will also inform other banks that are connected and are in cooperation with them such as botmaster mules' accounts. The banks will also send feedback reports to the affected bank in order to update the phishing detector which helps it to make a better decision for future attacks detection.

Step 11: The mules thereafter check whether money has been deposited to their accounts. They make withdrawal on individual basis if money has been deposited to their accounts, either direct from the bank or through Automatic Teller Machine (ATM), and thereafter remit the money to their botmaster. During this process or after their transactions, the appropriate law enforcement agents swing into full investigation using information provided to them by the bank. The law enforcement agents play a major role in recovering the stolen money back since their investigation is the last step of the framework.

The detection of attacks launched on victims account is the most vital step in the framework and it is largely depends on successful submission of honeytokens. This model of anti-phishing framework based on honeypot is shown in Figure 2.

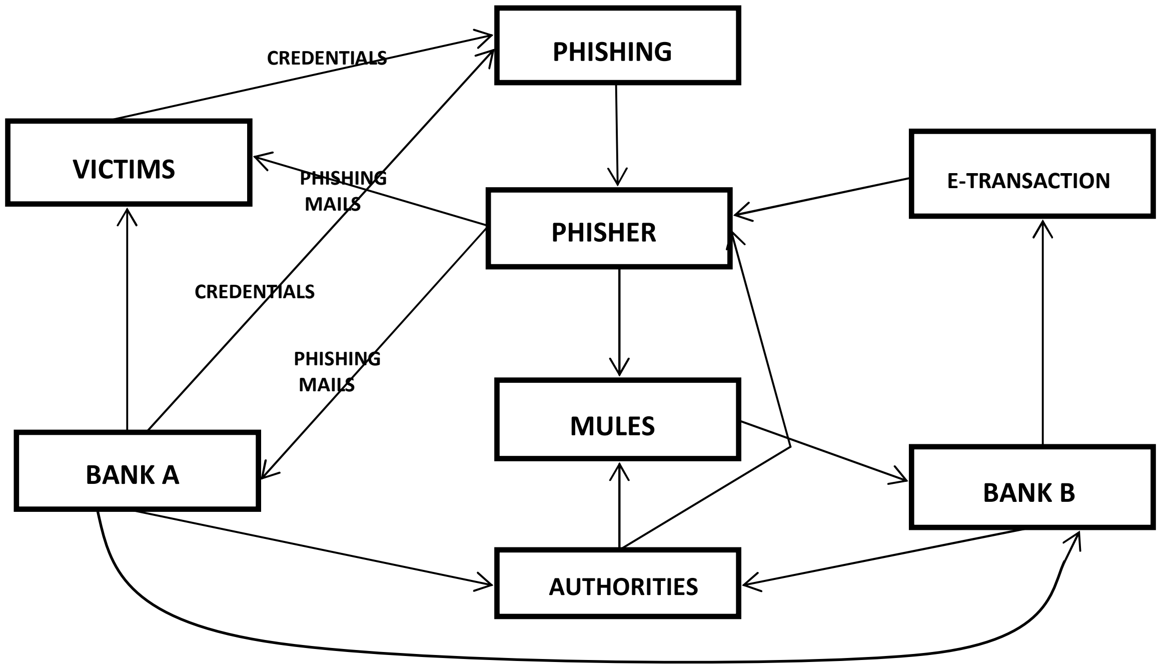

FIGURE 2: The novel anti-phishing framework based on honeypots [16].

[15] in this method make use of toolkits to submit honeytokens to attack website sites and uses honeybots such as Zeus to submit honeytokens to botnets, but it should be noted that they focused on the end-users or the money mules.

2.3 Limitations of the Existing Systems

[17] in their framework only targets to discredit a particular toolkit and the botmasters that created them. They fail to consider the proliferation of attack toolkits which is at increase in the labour market. [16] in their own model made use of toolkits to submit honeytokens to attack websites and uses honeybots such as Zeus to submit honeytokens to botnets. Though they tried to investigate different forms of attack but their targets were restricted to the end users, or money mules, without considering the actual toolkits that empowered them [17]

3. METHODOLOGY

The method adopted in this research is a process method. Starting with a review of the degree of attack and attack toolkits used to launch these attacks and mitigate them one after the other. The first step was to reach different antivirus companies especially Symantec antivirus company who has done a lot of researches on attack toolkits mitigation. Banks, individuals and other security forums were reached to gather information about the most widely used attack toolkits and the type of attack being launched using these toolkits. After the most frequently used attack toolkits were ascertained, it was reverse engineered. After which it was used to discredit the particular author that produce it as well as the end users.

Figure 3 illustrates the enhanced hybrid mitigation model with the following components - a honeyed e-banking system as a honeypot, honeytokens as fake or false identities created and managed by the e-banking system, attack toolkits for getting attack emails and sending honeytokens to attacked network or websites, honeybots for sending honeytokens to botmaster using attack toolkits. Honeytokens played a vital role in this model and they are used by both the honeybots and the banking system. An attack detector is placed in the banking system to automatically detect when an attack is taking place. We should emphasize that the banking system is not a pure honeypot since it also has all the functions of a normal e-banking system. It can be considered as a semi-honeypot. Figure 3.1 show the diagram of the proposed enhanced hybrid internet attack model, which when closely observed showcases the good work done by [17] and [16] with improvement over their limitations.

3.1 Stepwise Methodology Using the Architecture of the Enhanced Hybrid Mitigation Model

This sub-section gives the step by step attack toolkits mitigation method that is adopted in this research using the enhanced hybrid model. The hybrid model is an organized resolution encompassing different steps of the whole mitigation model adopted to mitigate the effect of attack toolkit in Nigeria commercial banking system and the model is shown in Figure 3.1.

Step 1: The financial institutions sets up a number of virtual machines running a number of honeybots and a number of spamtraps. Each spamtrap or honeybot is fed with a honeytoken. Following the normal characteristics of all bank customers, each honeybot accesses e-banking system from frequently and carryout virtual online transactions. A designated group of personnel in a bank take the responsibility of managing the virtual machines, spamtraps and honeybots. They also monitor doubtful phishing e-mails.

Step 2: The botmasters sends attack mails to prospective victims or infects victims' computers with attack software or attack toolkits, or mounts enhance attacks. The victims can be private or individual organizations, corporate organizations, multi-billion organizations, government of all levels, ministries and parastatals, and honeybot managers.

Step 3: The honeybot manager collects a lot of attack samples of mails from botmasters with the help of spamtraps. The honeybot manager equally monitors some anti-virus companies, other associative bank threat or attack records and other security forum to determine which attack is most paramount and which toolkit is mostly used to perpetrate the attack.

Step 4: The honeybot manager determines targeted websites URL fields by analyzing the identified toolkit using reverse engineering process.

Step 5: The honeybot manager generates false credentials or honeytokens which must appear real to botmasters, and incorporate it to the reverse-engineered toolkit to form honeybot or honey toolkit. These toolkits are also sold to the black market.

Steps 6: Each honeytoken should be submitted only once after which a new honeytoken will be created and assigned to attack toolkit by the group of managers after the old honeytoken is used up. Each toolkit should be updated after submitting a honeytoken to avoid it being detected.

Step 7: The botmaster collects some valid identities together with some false credentials from the attack website, or from the attack toolkit.

Step 8: The e-criminal visits the honeyed e-banking system with the credentials or honeytokens that have been collected to examine the validity of the honeytokens.

Step 9: The botmasters, mules or buyers of stolen credentials launches attack by trying to steal money from both the victims' accounts, and from the honeytokens.

Step 10: A detector placed in the honeyed e-banking system checks all e-transactions with respect to all false credentials or honeytokens. It sends a signal immediately it detects an attack by botmaster trying to steal a good or predefined amount of money from a honeytoken to a non-honeytoken account. The detector marks the receiver's account to be highly doubtful or suspicious which is in turn regarded as “phishing account”. This phishing account is usually a mule’s account opened and operated in another bank.

Step 11: The detecting bank requests for the consent of the prospective victims through SMS for reconfirmation of all money that are transferred from victims accounts to any other account that have being marked to be phishing accounts. The reconfirmation signal or message will be encrypted using RC6 enhanced encryption and decryption algorithm.

Step 12: The prospective victims then have the right to agree or refuse the transactions through the reconfirmation message. For the case of rejection, their account may be locked temporarily, and their credentials may be reset.

Steps 13: The detecting bank information appropriate authorities on the doubtful attack activities, it will also inform other banks that are connected and are in cooperation with them. The banks will also send feedback reports to the affected bank in order to update the phishing detector which helps it to make a better decision for future attacks detection.

Step 14: The mules check whether money has been deposited to their accounts. They make withdrawal on individual basis if money has been deposited to their accounts, either directly from the bank or through Automatic Teller Machine (ATM), and thereafter remit the money to their botmaster. During this process or after their transactions, the appropriate law enforcement agents swing into full investigation using information provided to them by the bank. The detection of attacks launched on victims account is another vital step in the framework and it is largely depends on successful submission of honeytokens.

Step 15: Public awareness and education to alert the potential victims about easy use of attack toolkits to steal their credentials over the internet and possible measures they can take to avert these threats.

Step 16: Enacting policies and law through the federal government and other anti-cybercrime bodies to fight tirelessly, the botmasters, mules and the black market dealers who make billions of money at the expense of the innocent victims.

Step 17: Internet Service providers (ISP) should put in place adequate internet security technologies to protect their potential customers from this epidemic security threat.

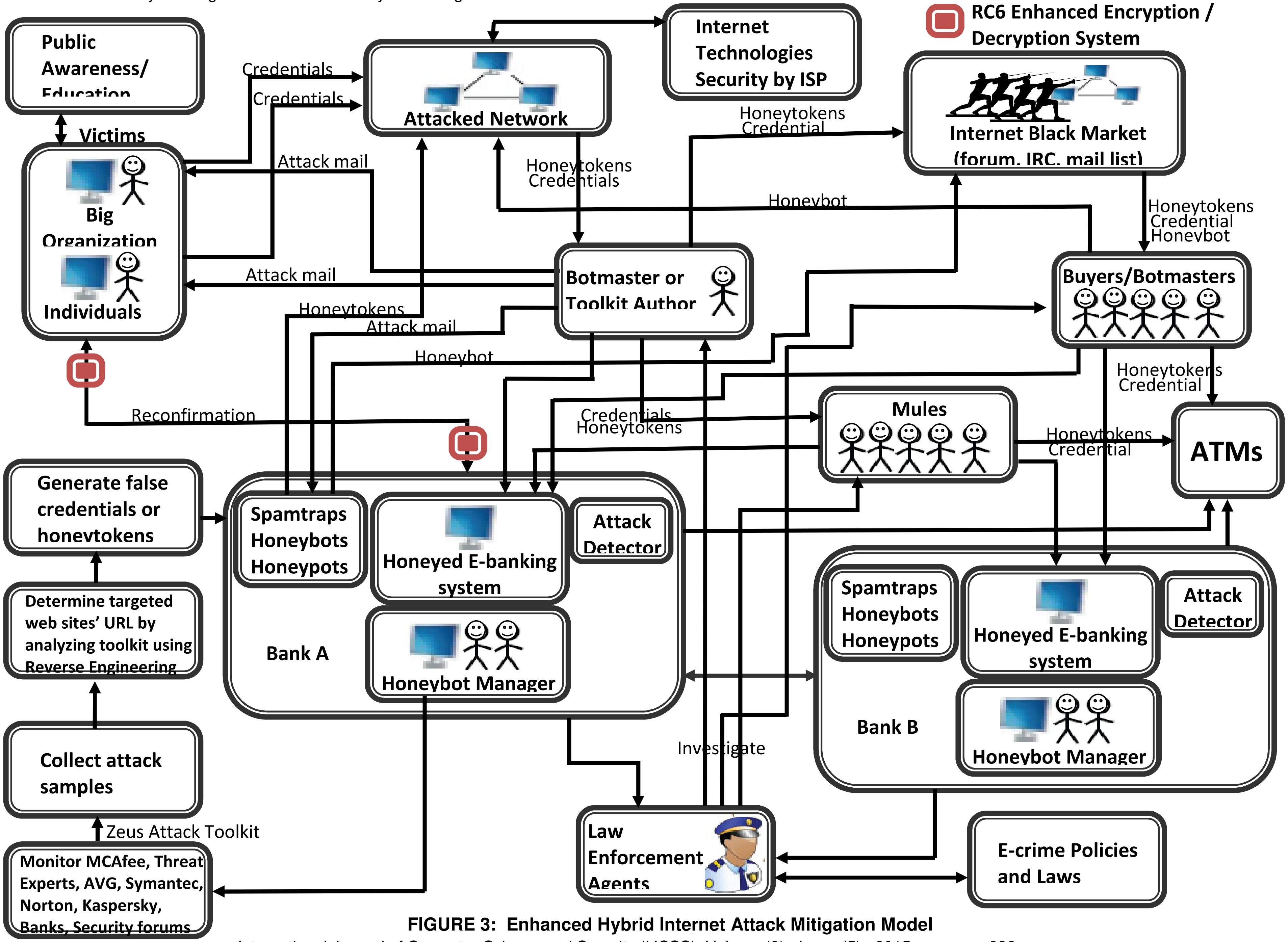
RC6 Enhanced Encryption / Decryption System
Public Awareness/ Education
Victims
Big Organization
Individuals
Credentials
Credentials
Attacked Network
Internet Technologies Security by ISP
Internet Black Market (forum, IRC, mail list)
Honeytokens Credential
Attack mail
Honeytokens Credentials
Honeybot
Honeytokens Credential Honeybot
Botmaster or Toolkit Author
Attack mail
Buyers/Botmasters
Honeytokens
Attack mail
Honeybot
Honeytokens Credential
Reconfirmation
Credentials Honeytokens
Mules
Honeytokens Credential
ATMs
Generate false credentials or honeytokens
Spamtraps Honeybots Honeypots
Honeyed E-banking system
Attack Detector
Determine targeted web sites' URL by analyzing toolkit using Reverse Engineering
Bank A
Honeybot Manager
Spamtraps Honeybots Honeypots
Honeyed E-banking system
Attack Detector
Bank B
Honeybot Manager
Collect attack samples
Investigate
Zeus Attack Toolkit
Monitor MCAfee, Threat Experts, AVG, Symantec, Norton, Kaspersky, Banks, Security forums
Law Enforcement Agents
E-crime Policies and Laws

FIGURE 3: Enhanced Hybrid Internet Attack Mitigation Model

3.2 Securing User Reconfirmation Using RC6 Encryption and Decryption Algorithm

[17] in their model proposed that the doubtful businesses or transactions should be reconfirmed at the bank by the bank staff, and not at the user end. Since the victim's system may have being compromised with attack toolkit, an out of band (OOB) medium can be used for the user reconfirmation process. They suggested that the OOB should include short message services (SMS), fax, telephone and postal service, the most convenient and practical medium should be SMS or telephone. With a known OOB medium, the user can carry out the reconfirmation process as follow: when the bank requests the user's consent through an OOB medium and ask the user to plainly reconfirm or refuse the doubtful transaction. The process went further to prohibit modification of the user's contact information in the banking; else the e-criminal that stays in between the two ends will use attack toolkit to utter it. The research equally propose the use of personal identification number (PIN) and transaction authentication number (TAN) to protect updating of the user contact information via telephone.

This existing proposed method of securing reconfirmation process using TAN is not reliable being that the man in the middle that is being avoided from internet medium also exist or have agents in our communication companies which TAN methodology chooses as alternative. The voice calls made from banks through communication medium can still be intercepted either physically by one of the attackers or different attacker working in that communication company, or technically via their broadband. The SMS also faces the same challenge. To enhance the security measure for securing the reconfirmation process as it is the ultimate last step for an attacker to be successful or failed. We proposed deployment of a cryptosystem which encrypt and decrypt the reconfirmation message to cipher text and back to plain text at both server or bank side and client or user side. The cryptosystem uses enhanced RC6 encryption and decryption algorithm together with private key to convert the reconfirmation message to cipher text which are random sequence of notations, symbols, numbers, alphabets etc, as the message propagate through the transmission medium to evade interception by attackers. The message is decrypted back to back to original plain text by the real user or bank official. It is the only legitimate owner who is in possession of the private key can decrypt the message. If the man in the middle intercepts the message, he will find it difficult to decrypt the message which is in turn useless to him. Again, the reconfirmation process should be timed and if there is no response at the end of the duration, the reconfirmation response should be considered negative. The schematic diagram of the cryptosystem is shown in Figure 4.

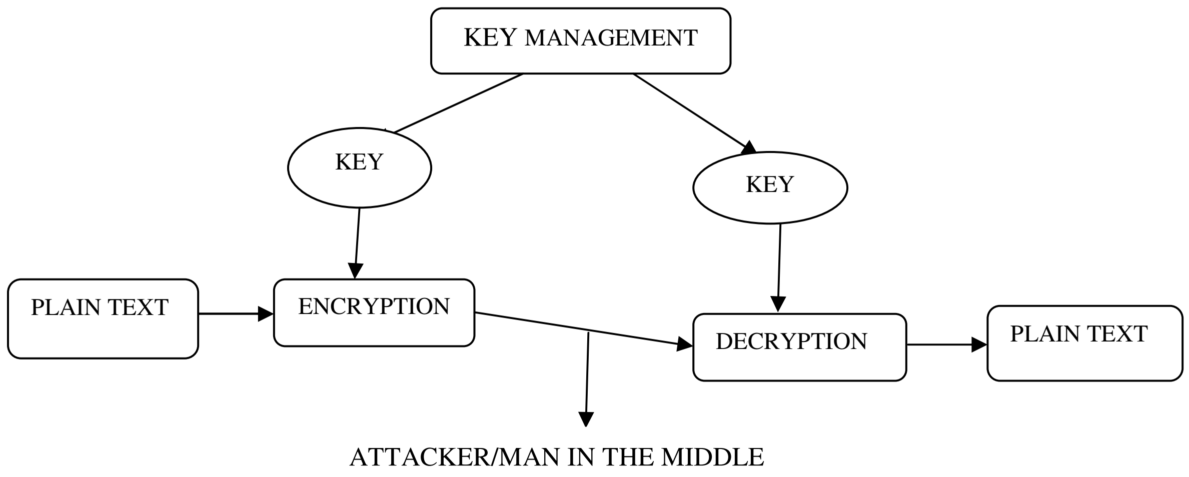

FIGURE 4: A schematic diagram of RC6 encryption and decryption system.

But the ultimate question here remains how do we transmit this secret or private key to avoid it been intercepted? This is been generated and confidentially sealed and handed over to the legitimate account owner directly at the point of creating the account.

4. DISCUSSION AND RESULT

The two subsystems; encryption subsystem and decryption decryption subsystem were tested individually. The result of the unit testing conforms to the set result. The two subsystems were integrated to form the main enhanced cryptosystem.

The testing process: The process of testing adopted in this research involve two major phases which conforms to verification and validation goals (V&V goals); verification ensures that the system conforms to its specification which implies that the system should encrypt/decrypt the reconfirmation messages as specified by enhanced RC6 algorithm; while validation ensures that the system does what the users expected of the system; that is encrypting the reconfirmation to cipher text and decrypting it to plaintext or original message without loss of confidence.

4.1 Test Data

Key 1

"It is just coded"

Plaintext 1

Dear Customer, this is to alert you that a transaction involving a cash transfer of two hundred million naira (N200m) from your account is currently going on. Please send confirmation to us if you actually wish the transaction to go successful. Thank you for banking with us and we look forward to serve you better. Management. ....

Cipher text 1

```
ØøE æjt w‡ ÿc   eN÷ ’K… FV   ¿Äß ý:Ñ ¸¢F ß¨—˜uM ]^ñÙU  ë¸šÖ_ÜèÂoû§ îå K1â QÛ„ zÑ ¶"Ý
÷æk küx Ko× È(9 ýšà ÉD› ×–Ì -ËÙ ‘µ žýy á>4 ý+   À%© [‹X H' ˜Ã1 [fj Ih‰ Jl -¿¤ ov> r²r PŸ `Ó~
*|¶ âë3 nëá ¿ ,‡A êØa yŒ€ 0™g…Ú p“– F ØøE æjt w‡ ÿc   eN÷ ’K… FV   ¿Äß ý:Ñ ¸¢Fß¨—˜uM
]^ñ ÙU   ë¸ šÖ_ ÜèÂ oû§ îå K1â QÛ„ zÑ¶"Ý ÷æk küx Ko× È(9 ýšà ÉD› ×–Ì -ËÙ ‘µ žýy á>4 ý+
À%© [‹X H' ˜Ã1 [fj Ih‰Jl -¿¤ ov> r²r  PŸ `Ó~ *|¶ âë3 nëá ¿ ,‡A êØa yŒ€ 0™g…Ú p“– F
```

Comparing the two results: From expected result/test case and actual test result, it was observed that the test data that is fed into the system for processing is the same as the result obtained after the encryption and decryption process, hence validation goal is achieved.

The result of this research is based on the justification of the new hybrid model. The term hybrid came into play as a result of the combination of the powerful features of the two models (that is, Botnet toolkit defamation process model and the novel anti-phishing framework based on honeypots). The new hybrid model targeted the most frequent used attack toolkits, looked into each one in turn as the most frequent used attack toolkit, perform reverse engineering on it and use it to discredit the particular author in charge of it. The model continues in that trend hunting all the powerful attack toolkits and discrediting them one after the other. The model also attacks the end users or the money mules. The model enhancement is as a result of its adoption of an enhanced Rivest Code version 6 (RC6) cryptosystem which plays a major role by guarding against the malicious manipulations of the man in the middle who is capable of intercepting the reconfirmation message. This is the last step to pull through the transaction if it is returned positive. The cryptosystem encrypts the reconfirmation message thereby hiding it from the attacker through the wireless transmission channel. RC6 cryptosystem is adopted in this research because it has been proved to be free from analytical attack which is also experimented in this research.

4.2 Comparative Evaluation of the current study showing its improvements over the Existing Related studies carried out by [16] and [17].

Existing researches' models by [16] and [17]	The current research hybrid model
• The research carried out by [17] damages the reputation of a particular botnet toolkit by destroying the confidence its users has on the toolkit. • It increases the insecurity of the toolkit users and make them liable for prosecution. • It reduces the profitability of the toolkit with respect to its use to sale credentials. • [16] concentrated on detection of attacks carried out on victims accounts. • [17] in their model proposed that the doubtful businesses or transactions should be reconfirmed at the bank by the bank staff. They suggested that an out of band (OOB) medium should be used for the user reconfirmation process which include short message services (SMS), fax, telephone and postal service,	• The hybrid model creates an opportunity to consider so many other types of botnet toolkits. • It destroys their reputations by making them insecure or unsafe to be used by their users. • The hybrid model creates opportunity to detect and prosecute the attacks launched on the victims' accounts. • It equally go further to root out individual forms of attack and reduces their effects to minimal. • The model also incorporates an enhanced security medium using enhanced RC6 cryptosystem. The OOB suggested by the previous researchers such SMS, fax, telephone and postal service, to be used for reconfirmation of transaction at bank side using PIN and TAN can easily be compromised by both internal attack (biased staff) and external attack (man in middle). • It also improvises e-crime policies and laws as tools to appropriate authorities if adopted will help to checkmate e-crimes. • It equally put in place measures to encourage ISPs to adopt sophisticated internet security technologies to checkmate e-crimes. • The hybrid also proposed public enlightenment campaigns as means of educating the public about the e-crime activities.

TABLE 1: Comparative evaluation of this study with the existing studies by [16] and [17].

4.3 Review of Achievements

This research establishes an enhanced hybrid attack toolkit mitigation model to discredit the toolkit authors. It equally establishes a cryptosystem using enhanced Rivest Code Version 6 (RC6) encryption and decryption algorithm to secure the reconfirmation message which played a very important role in the hybrid attack mitigation model. It proposes some effective policies and laws to the appropriate authorities which will empower the law enforcement agents to prosecute the cyber criminals adequately. One of the intellectual nuggets says that if one is not informed, he will be deformed. Hence, opening the eye of the potential victims to the reality of e-crime through public awareness will go a long way to help them to re-adjust their mentality and method of surfing the internet with security precautions. Therefore, this research also propose public lectures or awareness campaign to education the public which are the potential victims about cyber criminal activities, and also encourage the Internet service providers (ISPs) to adopt more sophisticated security technology measures to guard against cyber criminality and protect their potential customers.

5. CONCLUSION

Generally, talented e-criminals do not waste their precious time on attacks that will not profit them, and these smart attackers are the ones we are after in this research by putting in place various mitigation measures to discredit the toolkit. From the samples displayed in test data, it is obvious that the code is not transparent. Each letter of the plaintext does not have a fixed relationship to the cipher text. The number of characters in plaintext is 334 and its corresponding cipher text contains 453 characters which is higher. Since the corresponding number of characters in cipher text for every encrypted plaintext is not deterministic, it makes it very difficult to actually correlate a particular character or word in the plaintext to the cipher text.

6. SUGGESTION FOR FURTHER WORK

The implementation of other aspect of this hybrid model should be further looked into. It will be necessary to research for more behavioural methodologies that can be incorporated into this mitigation model. Also to design a measurement metric which will help to find out how much honeytokens are necessary to be injected into the internet black market in order to have major effect on botmasters, buyers and money moles.

7. REFERENCES

[1] M. Y. Rhee. Internet security: cryptographic principles, algorithms, and protocols. John Wiley & Sons Ltd., The Atrium, South Date, Chichestre, West Sussex. PO09 8SQ, England. 2003 Pp 26-298.

[2] F. Marc. “A white paper on Symantec Report on Attack Kits and Malicious Websites”, Symantec World Headquarters, 350 Ellis St. Mountain View, CA 94043 USA. Pp. 17-65. 2011. http://www.symantec.com/connect/blogs/zeus-king-underground-crimeware-toolkits

[3] Web source write-up on “Security Response from Symantec Corporation” 2010a. http://www.symantec.com/security_response/writeup.jsp?docid=2010-011016-3514-

[4] M. Chandrasekaran, R. Chinchani, & S. Upadhyaya. “Phoney: “Mimicking user response to detect phishing attacks”, In Proceedings of the 2006 International Symposium on the World of Wireless, Mobile and Multimedia Networks, pp. 5pp.–672. 2006.

[5] ITU Botnet Mitigation Toolkit Background Information, ICT Applications and Cyber security Division Policies and Strategies Department ITU. Pp. 12-43. 2008. Telecommunication Development Sector. www.itu.int/ITU-D/cyb/cybersecurity /projects/botnet.html

[6] Web source white paper on internet security threat report, from Symantec Corporation http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xv_04-2010.en-us.pdf;p.1831. 2010b.

[7] A. Ramachandran, N. Feamster, & D. Dagon “Revealing botnet membership using dnsbl counter-intelligence,” in Proceedings of USENIX SRUTI06, vol. 23. pp. 49–
54. 2006.

[8] T. F. Yen & M. K. Reiter. “Traffic aggregation for malware detection,” in Proceedings of the Fifth GI International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA08), pp. 207–227. 2008.

[9] G. Gu, R. Perdisci, J. Zhang, & W. Lee, “Botminer: Clustering analysis of network Traffic for protocol and structure independent botnet detection,” in Proceedings of the USENIX Security Symposium. Berkeley, CA, USA: USENIX Association. Vol. 31, pp. 139–154. 2008.

[10] G. Gu, P. Porras, V. Yegneswaran, M. Fong & W. Lee W. “Bothunter: detecting malware infection through ids-driven dialog correlation,” in SS’07: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium. Berkeley, CA, USA: USENIX Association. Vol. 19, pp. 1–16. 2007.

[11] J. Goebel, & T. Holz. “Rishi: Identify bot contaminated hosts by IRC nickname evaluation,” in Proceedings of USENIX HotBots07. Berkeley, CA, USA: USENIX Association. Vol. 32, pp. 7-25. 2007.

[12] J. Franklin, V. Paxson, A. Perrig & S. Savage. “An inquiry into the nature and causes of the wealth of internet miscreants,” in Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS’07), Vol. 37, pp. 375–388. 2007.

[13] T. Holz, M. Engelberth, & F. Freiling. “Learning more about the underground economy: A case-study of keyloggers and dropzones,” University of Mannheim, Tech. Rep. Reihe Informatik TR-2008-006, pp. 7-28. 2008.

[14] D. Birk, S. Gajek, F. Grobert, & A. R. Sadeghi,. “A forensic framework for tracing phishers, in IFIP Summer School on The Future of Identity in the Information Society”, Karlstad, Sweden, pp. 12-31. 2007.

[15] Z. Li, Q. Liao, & A. Striegel, “Botnet economics: Uncertainty matters,” in Proceedings of the 7th Workshop on the Economics of Information Security (WEIS’08), pp. 9-23. 2008.

[16] S. Li & R. Schmitz. “A novel anti-phisihng framework based on honeypots,” in Proceedings of the 4th annual Anti-Phishing Working Groups eCrime Researchers Summit. Vol. 16, pp. 3 - 38. 2009.

[17] Thomas Ormerod, Lingyu Wang, Mourad Debbabi, Amr Youssef, Hamad Binsalleeh, Amine Boukhtouta, & Prosenjit Sinha “Defaming Botnet Toolkits: A Bottom-Up Approach to Mitigating the Threat,” National Cyber-Forensics and Training Alliance Canada, Computer Security Laboratory, Concordia University, Montreal, Quebec, Canada, H3G 2W1, 2010.

OpenGL Based Testing Tool Architecture for Exascale Computing

Muhammad Usman Ashraf *m.usmanashraf@yahoo.com*
Faculty of Information and Computer Technology
Department of Computer Science
King Abdulaziz University
Jeddah, 21577, Saudi Arabia

Fathy Elbouraey Eassa *fathy55@yahoo.com*
Faculty of Information and Computer Technology
Department of Computer Science
King Abdulaziz University
Jeddah, 21577, Saudi Arabia

Abstract

In next decade, for exascale high computing power and speed, new high performance computing (HPC) architectures, algorithms and corrections in existing technologies are expected. In order to achieve HPC parallelism is becoming a core emphasizing point. Keeping in view the advantages of parallelism, GPU is a unit that provides the better performance to achieve HPC in exascale computing system. So far, many programming models have been introduced to program GPU like CUDA, OpenGL, and OpenCL etc. and still there are number of limitations for these models that are required a deep glance to fix them. In order to enhance the performance in GPU programming in OpenGL, we have proposed an OpenGL based testing tool architecture for exascale computing system. This testing architecture detects the errors from OpenGL code and enforce to write the code in accurate way.

Keywords: Exascale Computing, OpenGL, OpenGL Shading Language, GPU, CUDA, Parallelism, Exaflops.

1. INTRODUCTION

This guideline is used for all journals. These are the manuscript preparation guidelines used as a In computing system, Exascale brings up to a computing technology that has ability to achieve the performance in excess to one exaflop [6]. The current technology has capability of performance in petaflops range. The object to enhance the computing performance presents a number challenges at both software and hardware levels. To make possible to targeted flops, many computer companies are working on both software and hardware level to increase the computing performance by implementing on-chip parallelism.

Since last decade, Graphics Processing Unit (GPU) based parallel computing technologies have brought up an extensive popularity for high performance (HPC). These technologies including Compute Unified Device Architecture (CUDA), OpenGL, and OpenMP etc. have opened many new research directions and visions for future Exascale computing system. However, at the programming level there are still a number of major challenges that should be fixed by introducing the new algorithms and architectures [4].

In this paper, we have emphasized on OpenGL software interface to program a GPU. Basically, before GPU processing, there is a pipeline having number of stages. Some of those are programmable and some are non-programmable or configurable stage in pipeline [3]. Our focus is on programmable steps using OpenGL Shading Language that are discussed in section II.

Further, writing program in GLSL we found number of common error those occurred at programming level and on the bases of these found errors we proposed a testing tool architecture to increase the performance and accuracy in program. This proposed architecture provides guarantee that there will be no error in the code once a developer follow this specific technique.

Further, rest of the paper is organized in such a way that, section II consist of briefly description of basic architecture of a system with GPU presence. Section III describes the classification of programmable and non-programmable stages through pipeline in OpenGL. In section IV we have explained the number of errors found during writing program in GLSL with help of code. In section V, we have presented a testing tool architecture and its behavior. Section VI consist the tool architecture evaluation description and then conclusion and future directions is presented in last section VII.

2. GPU WITHIN MACHINE

This guideline is used for all journals. These are the manuscript preparation guidelines used as a In modern computers, GPU acts as a second computer. Similar to CPU It also has its own memory and processors. The CPU get input from user and classified either it's related to GPU or CPU itself for processing. In case of GPU processing, it forward the information to GPU for processing through a programming language like OpenGL, OpenCL or CUDA. GPU process the task using its own RAM and send the processed information back to CPU for further utilizing [2].

However a GPU is designed to perform multiple kinds of tasks including graphics rendering, geometric calculations, complex mathematical calculations etc. Moreover, GPU is a basic unit for parallel processing to accomplish high performance computing. A basic architecture is presented in figure1.1 showing how GPU is resided in a computer system.

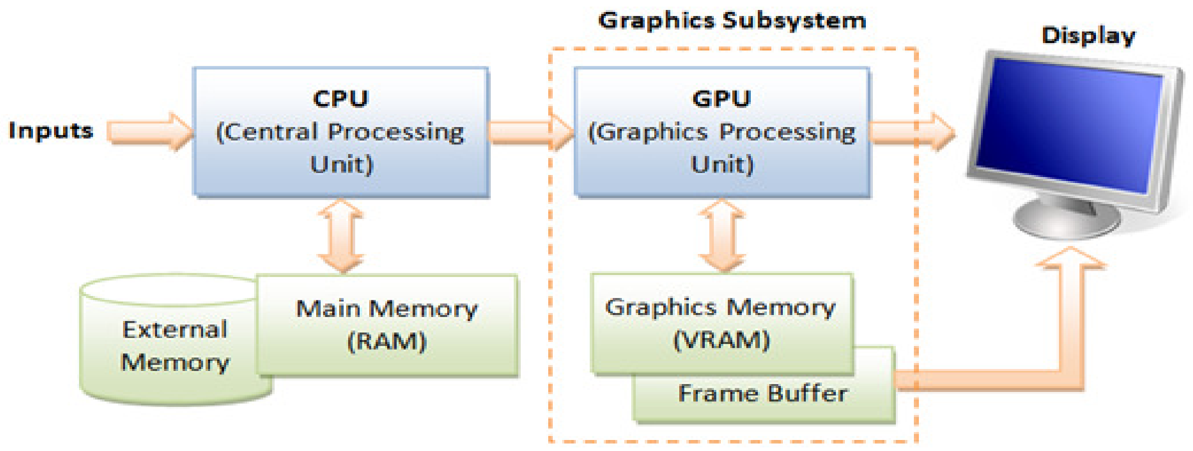

FIGURE 1: Computer system with GPU.

Figure 1 clearly showing that in a computer system CPU and GPU are almost similar but having different kind of operation processing.

3. GRAPHICS RENDERING PIPELINE

A pipeline consists of number of steps that are processed in series in such a way that one stage gets an input and provides output to next stage. Similar to this computing terminology, graphics rendering pipeline also behaves in same way as shown in figure2. This rendering pipeline gets the raw vertices and primitives as an input in 3D form. Vertex processing is the first step that gets this raw data, processes it and sends transformed vertices & primitive's data as input to rasterizer. Rasterizer further process data and convert each primitive into set of fragments. Further, fragment processor process each fragment by adding colors, positions, normal values etc. and generate as input data for output merging step. Continuing, the pipeline steps, output merging aggregates the fragments of all primitives from 3D to 2D shapes for the display. Finally, the output object is usable for GPU as input data. In modern GPU, the pipeline stages are categorized into two types as follows:

- Programmable
- Non-Programmable.

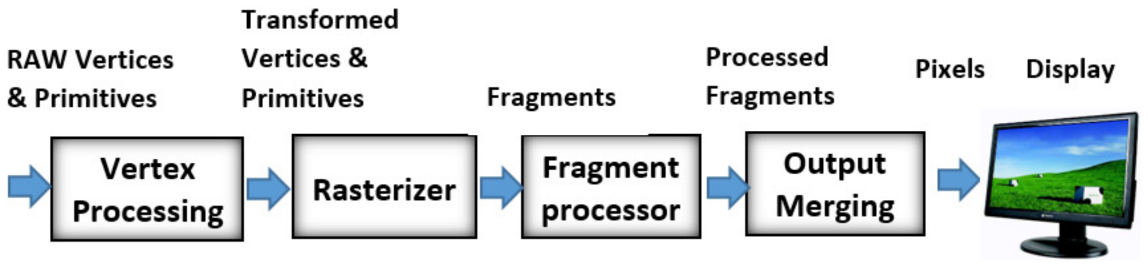

FIGURE 2: Graphics Rendering Pipeline [5].

Vertex processing and fragment processing are programmable stages from the graphics rendering pipeline and rasterization and output-merging are the stages that are non-programmable or configurable by using the GPU commands. In programmable stages, we need to write the program for vertices and fragments know as vertex shader and fragment shader respectively. Different programming models (C for graphics, high level shading language (HLSL), OpenGL Shading language (GLSL) etc.) have been introduced to write these shader programs [8]. In section IV, we have discussed how to program these shaders using GLSL and their perspective type of errors during writing code [5].

4. GLSL PROGRAM AND TYPES OF ERRORS

4.1 OpenGL Shading Language

The OpenGL Shading Language (GLSL) is the principle shading language for OpenGL. OpenGL provides many shading languages but GLSL is the one which is closer and the part of OpenGL core. GLSL is very popular programming model to write the shaders for GPU but it is very important to understand the graphics pipeline and the sequence of shader creating before writing in any language [1]. There are some particular steps to write a shader as follows:

- Create Shaders (creating new shader object)
- Specify Shaders (load shader source)
- Compiling Shaders (Actually compile source)
- Program setup (creating shaders object)
- Attach Shaders to Programs
- Pre-linking (setting some parameters)
- Link Shaders to Programs
- Cleanup (detach and delete all shader objects)
- Attach program to its stage
- Finally, let GPU know Shaders are ready.

4.2 Types of Error

During writing the vertex and fragment shaders by following above steps in GLSL there are variety of errors that occurs in a normal program. So, it is very important to have a deep look at those kind errors including how these errors occurs and reason of occurrence. These errors are as follows:

4.2.1. *Error in Vertex Stream data*
Error in vertex stream data is the type of error that normally occurs at very initial stage of writing the program when we try to get vertex buffer array (VBA). From GLSL side, a file name is passed that's buffer array is required. Before return the buffer array data of file there might be possibility of errors as follows:

- File doesn't exist at given path
- Accessed file is corrupted file
- Forgot to write method for reading the file

4.2.2. *Compilation Error*
On shader compilation, it must be checked that the compilation process of shader is successfully completed or not. Once compilation is completed, it returns the compilation status weather it is successfully compiled or not. In case of false there will be a compilation that should be handled from before moving to next stage.

4.2.3. *Missing vertex and fragment necessary parameters*
There are some necessary parameters for fragment and vertex that must be passed by accurate values in calling specific methods from GLSL:

glBindAttribLocation(vertexProgram , 0, "**Position**");
glBindFragDataLocation(geomFragProgram, 0, "**FragColor**");

Once these parameters are missed or wrong values are entered, there will be error occurred on compilation of these methods.

4.2.4. *Linking Error*
Similar to compilation, linking process can also be failed due to passing faulty programming for linking or some internal linking processing. There should be validation of linking process by returned value either its false or true mean linking process is successfully completed or not.

4.2.5. *Missing Program Attachment with Perspective Stage*
Another step during shader programming is attaching program with its perspective stage. In case, attachment process is skipped; there will be an error occurred in the program. Normally below methods is used for attachment.

glUseProgramStages(pipeline, VERTEX_BIT, vertxprogrm);

4.2.6. *Uniform Count Error*
In GLSL, a uniform is global variable that is declared with the keyword as "uniform" storage qualifier. These act as parameters that the user of a shader program can pass to that program. To pass the parameters following methods is used in GLSL:

glUniform2fv(lightposHandle, amount, lightpositions);

Here, vertex array length which should always be same as value as per array length. Dissimilar value will be cause of error in the program.

4.2.7. *Final Program Validation*
Finally, before using the program in GPU, the program validation is necessary. Program validation will tell us, the program is acceptable for GPU or not. Normally, this happen, a program is used for GPU without validation and due to error in program whole application crashed. Below method is used to validate the program:

glGetProgramiv(programme,GL_VALID_STATUS, &params);

Valid_Status parameter returns the status of program in true/false either the program is valid or not.

5. PROPOSED TESTING TOOL ARCHITECTURE

In last section we have discussed the number of challenges that might be occurred during writing the shader program using OpenGL Shading Language (GLSL). These errors in program will affect the system performance and efficiency. In order to avoid from these specific error, we have proposed testing tool architecture. By following this architecture, we can write an error free code for shading program to use in GPU. As in GLSL, there are number of steps to program a shader. We divided all the steps in four major categories as shown in figure3.

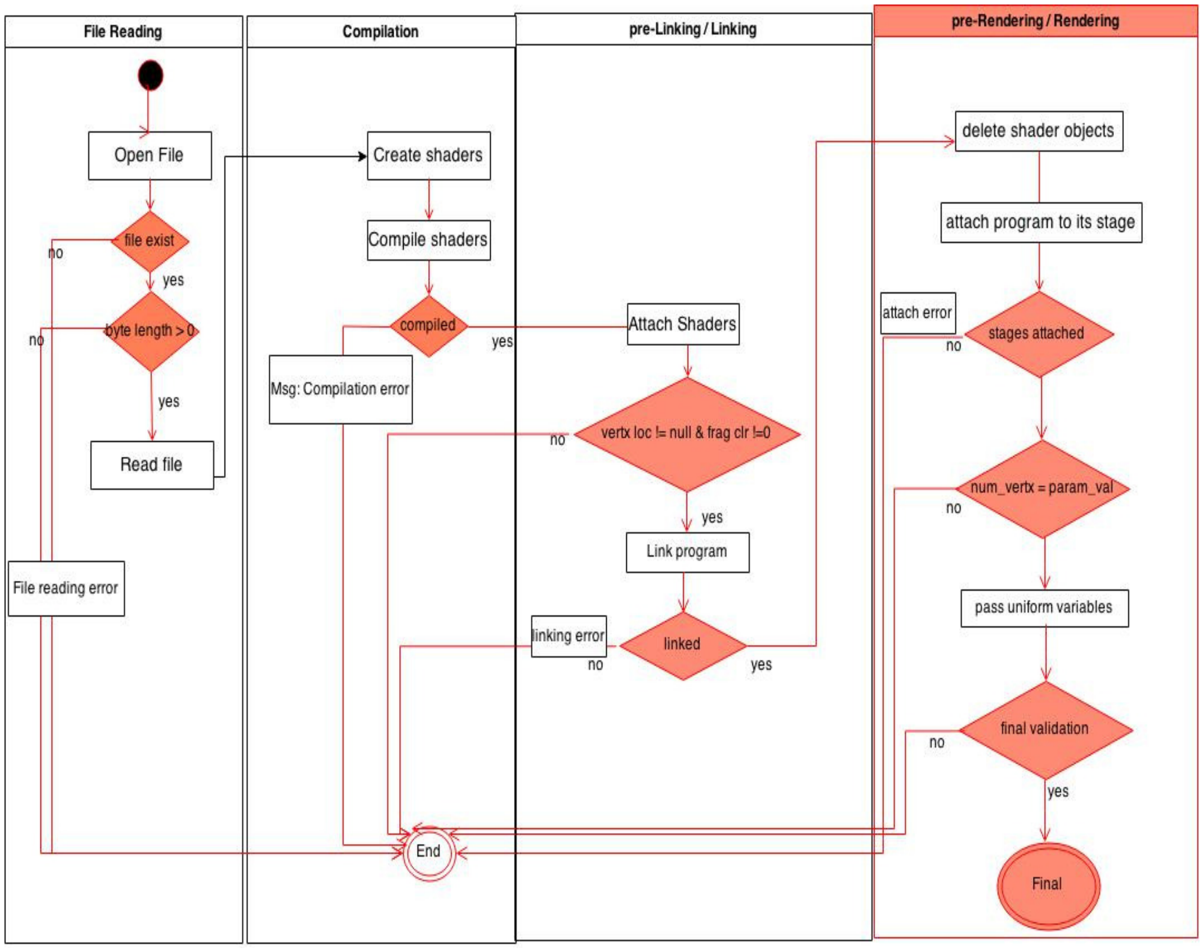

FIGURE 3: Testing Tool Architecture.

5.1. File Reading

The very first step is to read the file which's buffer array is required. In this section, our architecture validates some operations on that particular file to make sure that does the file exist at given path, does the file is valid and does the file read before passing to shader as a buffer array. If the file doesn't exist at give path or the accessing file is corrupted, the architecture imposes to terminate the program by writing the else cases and avoid to crash the program.

5.2. Compilation

In next section, after compilation it imposes to check the compilation results weather the compilation is completed successfully or not. In case of any error in compilation, the architecture enforces to terminate the program to avoid further crash in application.

5.3. Pre-Linking and Linking

After successful compilation, next step is validation of before and after linking the program. Before linking, some necessary correct parameters which are required for both vertex and fragment as well. The proposed architecture insists to pass the accurate required parameters such as position is required for vertex shader and colors for fragment. After linking, similar to compilation, it imposes to check that linking process is successfully completed or not and take action accordingly.

5.4. Rendering

Rendering step consist of many validations such as program attachment with its perspective stage in pipeline. According to our architecture, the programmer must write the GLSL statement which is used to attach the program with a specific stage. Conversely if this attachment is skipped, the program will unable to find its stage and application will be crashed.

Another type of error is related to uniform global variable as parameter value that passed for vertex specifically. glUniform2fv is method used that get three parameters in it. One of those is the length of vertex array which is created with uniform keyword. If the passing length value is dissimilar, error will be occurred in program and crashed. The last validation step in rendering section is validation of final program. Final program must be validated before using in GPU to make sure there is no error in the created program.

6. TESTING TOOL ARCHITECTURE EVALUATION

This testing tool architecture for OpenGL is proposed basically to improve exascale computing system. GPU is the basic unit that is being used to enhance the power of a system to achieve exascale computing. However, In order to achieve this certain level performance OpenGL play a major role to program graphical processing unit. This architecture helps us to detect the possible number of errors from the code written in OpenGL and improve the processing power of code as well. Keeping in view the program structure of an OpenGL Shading language, we have proposed the testing architecture to evaluation the number errors in our code that could be cause to decrease the performance of a system. Using this proposed testing tool architecture, we can evaluate our code written in GLSL and make error free by following it.

7. CONCLUSION

High performance computing architecture is the vision of next decade for exascale system. Many new technologies, algorithms and techniques are required to achieve exaflop computing power. In modern computers, GPU is basic unit that can provide the required performance for exascale system. So far, there are still many limitations for existing technologies as CUDA, OpenGL, and OpenCL etc. to program such a GPU unit. In order to enhance the performance of OpenGL code, we have presented a testing tool architecture in this paper. This proposed architecture insists the developers to write an accurate code by following the structure of proposed architecture. By future perspective, this proposed architecture will help to achieve high performance computing using OpenGL for exascale computing system. Still there are many open challenges for GPU in different technologies to develop an exascale computing system.

8. REFERENCES

[1]. J. Kessenich and D. Baldwin, “The OpenGL Shading Language”, Sep 2006 .

[2]. D. Luebke and G. Humphreys, “How GPU works”. Feb 2007.

[3]. M. J. Kilgard and J. Bolz, “GPU-accelerated Path Rendering”, Computer Graphics Proceedings, Annual Conference Series. 2012.

[4]. “Exascale computing research.” Internet: http://www.exascale-computing.eu/presentation-2/, [May. 3, 2015].

[5]. “3D Graphics with OpenGL Basic Theory” http://www.ntu.edu.sg/home/ehchua/programming/opengl/ cg_basicstheory.html, July. 2012 [April, 14. 2015].

[6]. “GLSL Tutorial Core” http://www.lighthouse3d.com/opengl/glsl/, June, 23. 2014 [May. 10, 2015].

[7]. D. Shreiner, “Performance OpenGL: Platform Independent Techniques”, SIGGRAPH. 2001.

[8]. S.F. Hsiao, P. Wu, C.W. Sheng, and L.Y. Chen, “Design of a Programmable Vertex Processor in OpenGL ES 2.0 Mobile Graphics Processing Units”. IEEE. 2013.

Hybrid Model Based Testing Tool Architecture for Exascale Computing System

Muhammad Usman Ashraf *m.usmanashraf@yahoo.com*
Faculty of Information and Computer Technology
Department of Computer Science
King Abdulaziz University
Jeddah, 21577, Saudi Arabia

Fathy Elbouraey Eassa *fathy55@yahoo.com*
Faculty of Information and Computer Technology
Department of Computer Science
King Abdulaziz University
Jeddah, 21577, Saudi Arabia

Abstract

Exascale computing refers to a computing system which is capable to at least one exaflop in next couple of years. Many new programming models, architectures and algorithms have been introduced to attain the objective for exascale computing system. The primary objective is to enhance the system performance. In modern/super computers, GPU is being used to attain the high computing performance. However, it's the objective of proposed technologies and programming models is almost same to make the GPU more powerful. But these technologies are still facing the number of challenges including parallelism, scale and complexity and also many more that must be fixed to achieve make computing system more powerful and efficient. In this paper, we have present a testing tool architecture for a parallel programming approach using two programming models as CUDA and OpenMP. Both CUDA and OpenMP could be used to program shared memory and GPU cores. The object of this architecture is to identify the static errors in the program that occurred during writing the code and cause absence of parallelism. Our architecture enforces the developers to write the feasible code through we can avoid from the essential errors in the program and run successfully.

Keywords: Exascale Computing, GPU, CUDA, OpenMP, Parallelism, High Performance Computing (HPC).

1. INTRODUCTION

High Performance Computing (HPC) technology architectures and algorithms are anticipated to transfer dramatically in the future. Accordingly, increasing on-chip parallelism is becoming the objective of computer companies to achieve high performance [1]. In modern computers, Multi-core technology is proving very good offer in order to get high performance and power efficiency. With perspective of programming model, in order to take advantage of multi core technologies architecture, OpenMP was introduced [5]. Another major challenge for exascale computing is to parallelism in computing.

1.1. CUDA Programming

CUDA (Compute Unified Device Architecture) is a parallel computing architecture developed by NVIDIA [2]. For this purpose, usage of GPU is introduced that consist of multi core resided in it. GPU is similar to CPU in a computer system but is very powerful. Many programming models are available to write program for GPU but CUDA by NVIDIA is the best option in order to achieve parallelism through GPU processing [8]. CUDA provides variety of key abstractions shared memory, parallelism in computing for multi cores and barrier synchronization as well. Moreover, CUDA architecture provides strong computational interfaces including OpenGL and Direct Compute [9]. CUDA programming model overcome the challenges that are face in parallelism.

1.2. OpenMP Programming

Open multi-processing OpenMP is a programming model that have capability to handle multithreading by computing in parallel module. The basic idea behind this programming model is data processing parallelly. OpenMP consists of number of directives and libraries that are called runtime [2]. It also processes the looping region as parallelized by inserting compiler directives in starting region of OpenMP module that makes the program more efficient and improves overall application performance [3]. An example of parallelism in loop region using OpenMP is show as follows:

```
# pragma omp parallel shared(a1,a2,a3,chunk) private(i)
{
#pragma omp for schedule (dynamic, chunk) nowait
for (i=0; i < N; i++) {
a3[i] = a1[i] + a2[i];
}
}
```

1.3. Hybrid Programming

In this paper, we have used a hybrid programming model by combining CUDA and OpenMP as well. The purpose to use both models at a time is to write program for of GPU and shared memory [8]. However we can write program for GPU in order to make parallel processing in both GPU and CPU as well. As CUDA will handle GPU parallelism and OpenMP to run the CPU process in parallel way [6].

1.4. GPU Architecture

In modern computers, GPU acts as a second computer. Similar to CPU It also has its own memory and processors. The CPU get input from user and classified either it is related to GPU or CPU itself for processing. In case of GPU processing, the information is forwarded to GPU for processing. GPU process the task and send the processed information back to CPU for further utilizing [12]. GPU consist of two main components.

- Global memory
- Streaming Multiprocessors (SMs)

Global memory is accessible by both GPU and CPU inside the system. Second component streaming multiprocessors is the core part of GPU that performs the actual computation for GPU [12]. SMs consist of multiple cores and shared memory where each core has its own control unit (CU), registers, execution pipeline and cashes. The basic architecture of a GPU is as follows:

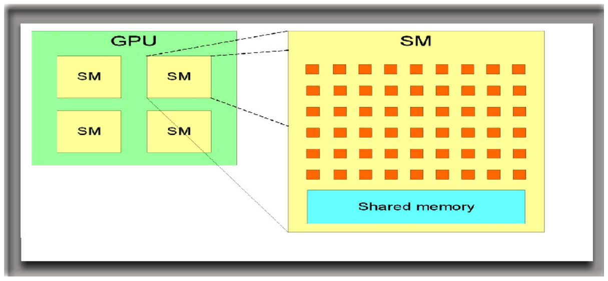

FIGURE 1: GPU Basic Architecture.

Rest of the paper is organized in such a way that, Section II describes the type of errors that are the hurdle to accomplish the GPU goal using CUDA and OpenMP. Section III deals with the errors discussed in section II and also solution to avoid such sort of hurdles in program. Section III presents the testing tool architecture that prevents from the type of errors that are discussed in section II. Further, a conclusion is presented in section IV.

2. TYPE OF ERRORS IN HYBRID MODEL

In this section, we have presented different type of errors that occurred during writing a program for GPU cores and shared memory using hybrid programming model. Let's discuss these errors one by one explaining how to produce in program.

2.1. Data Race

It is basically a computational hazard that comes up when the results of the program depend on the execution for another program. Such kind of error arises normally when two or more threads are running in parallel [6][7]. A simple scenario how data race error occurs in program is described in below example.

Let's x that points to a shared memory in a program. The requirement is to get the increment in x value. In sequential processing, this normally happen in three steps as:

1) Read the value of x into a register
2) Add 1 to the value read in step 1.
3) Write the results back to x and final results.

But what about parallel processing of same scenario, race condition produced or not? Let's discuss the same problem in parallel processing where multiple threads are present and process parallelly.

Assume that initial value of x is 3

1) Thread A reads the value 3 from x.
2) Thread B reads the value 3 from x.
3) Thread A adds 1 to its value 3, to make 4.
4) Thread A writes its value 4 back to x.
5) Thread B adds 1 to its value 3, to make 4.
6) Thread B writes its value 4 back to x.
7) Final value in x is 4 which is incorrect.

Why this happen even this was not to complex statement? Does it due to data race? Answer is yes but how we can avoid occurring such error in hybrid programming model.

Solution:

In CUDA, we can avoid from occurrence data race problem by adding some additional statements [11]. Normally Locking / Unlocking and atomic operation are used to handle data race error in CUDA. These operations process the threads in traditional sequential way.

Lock / Unlock operation

```
global void  test1( Lock lock , int  *nblocks)
{
        i f ( threadIdx.x == 0 )
        {
                lock.lock( ) ;
                * nblocks = *nblocks + 1 ;
                lock.unlock ( ) ;
        }
}
```

In above example code, the statement written inside lock and unlock block will be executed sequentially. In this way, there will be no data race but obviously the performance is affected.

Atomic Operation in CUDA

```
#include <stdio.h>
#include<cuda.h >
global   void   colonel ( int *b_f ) {
        atomicAdd ( b_f , 1 ) ;
}

int main ( )
{
int a = 0 , *a _d ;
colonel <<<1000,1000>>>(b_f ) ;
}
```

In above example code, atomic statement has been used to avoid from data race problem writing code in CUDA. Atomic is built in method used in CUDA for also other operation like: multiplication, subtraction, division etc.

Atomic Operation in OpenMP
Atomic operation is also used in OpenMP for same purpose but here atomic keyword is used as directive in the code. Adding this directive the code written inside the OpenMP block will be processed sequentially.

```
#include <omp.h>
int count;
void Tick()
{
  #pragma omp atomic
     count = count+1;
 }
```

In above code example atomic is the directive that specifically used for running the code sequentially inside it.

2.2. Use of lock without Unlocking
This is error related to using locking/unlocking statement in the program. As we have discussed the usage of lock and unlock statement to avoid data race condition in program, however it is also necessary to make sure that unlock statement is also present there once lock statement is used in the program [4].

2.3. Use of ordered clause without ordered construct
Another type of error occurred normally in OpenMP part from hybrid programming model which is the usage of ordered clause with ordered construct. Once an ordered clause is placed within for work-sharing construct and developer forgot to place a separate ordered clause inside for loop, an error will be occurred in the program and application will be crashed.

2.4. Use of critical when atomic operation is sufficient
Basically, this is performance related issue that normally considered when we use critical clause in the program even the problem could be solve by using simply atomic clause rather than critical [4]. By usage of critical clause in this scenario, the overall system performance will be affected.

2.5. Placing much code inside Critical region
Normally it has been seen that the novice programmer places a lot of code in the critical section and cause the number of errors. These errors could be cause of further two sub errors as follows:

- Blocking the other threads more than required.

- Paying the maintenance cost more than expected associated with that specific region [4].

Example code:
```
#pragma omp parallel for
for ( i = 0 ; i < N; ++i )
{
#pragma omp critical
{
        if ( arr [ i ] > val)
  {
        if ( arr [ i ] > max)
  max = arr [ i ]
  .
  .
  .
  .
  and a lot of more statements…
  }
}
```

So, it is suggested to resort reduce the code written inside the critical clause region.

2.6. Missing for in "#pragma omp parallel for"

It is very common type of error in OpenMP, mostly people use 'for' clause to include a loop statement inside the '#pragma' region but forgot to add 'for'. This error leads every thread the whole loop, but not only parts of it [4].

Example code:
```
        #pragma omp parallel for
        {
                if ( arr [ 0 ] > val_1)
                {
                        if ( arr [ 1 ] > val_2)
                        max = arr [ 2 ] ;
                }
        }
```

In above code, as 'for' clause is being used but no for is used inside the region. It will cause of error occurrence in the program.

3. PROPOSED ARCHITECTURES AGAINST DETECTED ERRORS

In order to avoid the errors that we have discussed in the last section, we have proposed the testing architectures for particular errors that must followed by the developer before/during writing code.

3.1 Data Race

As data race is the common error that occurs normally when we write program for GPU cores and shared memory as well. Basically, data race is type of error that occurs in a program when the same statement is required for two or more different process. Each process in a program want to compute its set of statements as soon as possible. But this could only be possible if all required statements are in hand and could be executed in parallel. In case of availability of all the parameters and statements that specific process could be executed in parallel otherwise a race condition will be occurred. It means that the same data is being used in any other statement or waiting for some result from other process. So, to avoid this issue in parallel processing, firstly we analyze the code and the resolve the issue on behalf of condition. One of the solution is to compute that specific part of code as sequentially using locking/unlocking statements.

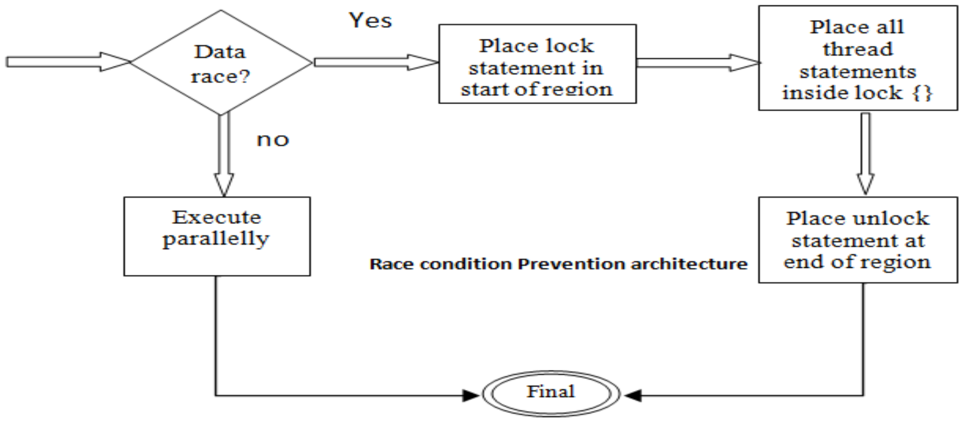
Data race?
Yes
Place lock statement in start of region
Place all thread statements inside lock {}
no
Execute parallelly
Race condition Prevention architecture
Place unlock statement at end of region
Final

FIGURE 2: Data Race prevention architecture for CUDA.

The above architecture is related to error for race condition (A) in CUDA and also the prevention of errors that occurs by adding locking (B) statement in the program.

Similarly, for data race problem in OpenMP, once you have analyzed the code to find out that either there is data race exist or not. After that you should follow the below proposed architecture that will ensure you to write error free code.

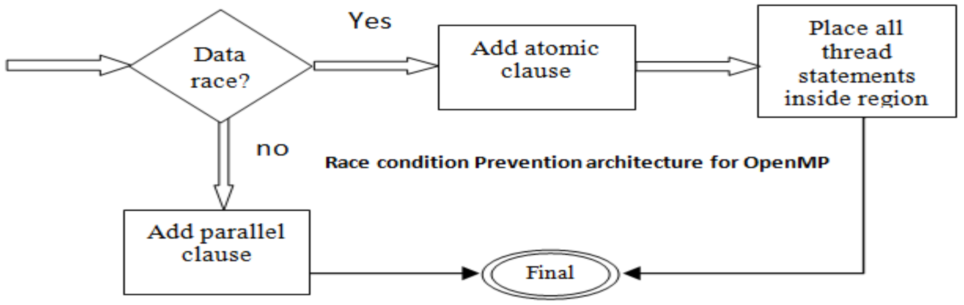
Data race?
Yes
Add atomic clause
Place all thread statements inside region
no
Race condition Prevention architecture for OpenMP
Add parallel clause
Final

FIGURE 3: Data Race prevention architecture for OpenMP.

3.2 Missing "For" keyword when using "For" Clause

Another architecture is presented to handle the error that occurs due to missing 'for' when 'for' clause is used in the program. This error is related to OpenMP from hybrid programming model. Below is the diagram representing that how we can insist a developer to must add 'for' when for clause has been used with '#pragma' statement.

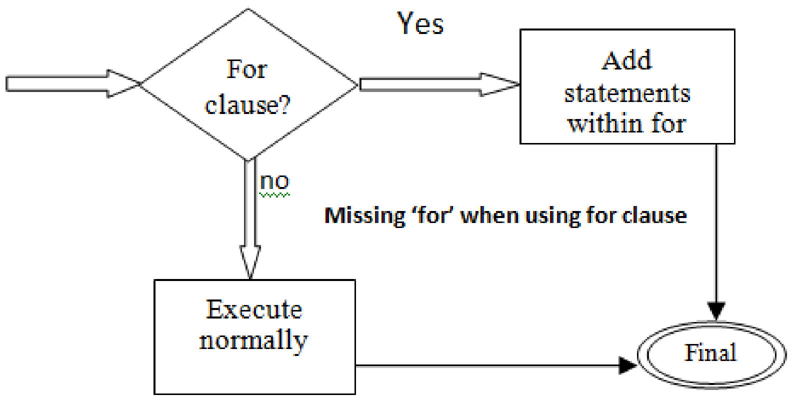
For clause?
Yes
Add statements within for
no
Missing 'for' when using for clause
Execute normally
Final

FIGURE 4: missing 'for' when using 'for' clause .

In previous section, rests of the errors are performance related, that must also be avoided to occurrence in the program. To handle these errors, the prevention statements should be followed before writing the program.

4. TESTING TOOL ARCHITECTURE EVALUATION

This testing tool architecture for hybrid model is proposed basically to improve exascale computing system. GPU is the basic unit that is being used to enhance the power of a system to achieve exascale computing. However, In order to achieve this certain level performance, CUDA, OpenMP do play a major role to program graphical processing unit. This architecture helps us to detect the possible number of errors from the code written in CUDA and OpenMP and improve the processing power of code as well. Keeping in view the programming layout in both the languages, we have proposed the testing architecture to evaluation the number errors in our code that could be cause to decrease the performance of a system. Using this proposed testing tool architecture, we can evaluate our code written in hybrid model languages such as CUDA and OpenMP and make error free by following it.

5. CONCLUSION

This paper is presented to make enhancement in exascale computing system. In order to obtain this objective, GPU which is the core unit for exascale system should be reviewed deeply to make more powerful. In this paper, we emphasized on parallelism and shared memory utilization in parallel and presented hybrid model based testing tool architecture for exascale computing system. In hybrid model, we add two CUDA and OpenMP programming models to enhance the system performance. We presented some major type of errors when a developer writes the program for GPU using this hybrid programming model. Further we discussed each error in detail by specifying that how these errors occurs in the program what is the cause of occurrence and how we can avoid from these errors during writing the program. We presented different architectures which specify that how to avoid these errors. Our architectures ensure to a developer to write an error free code if this is followed properly.

For future perspective, there is still need to emphasize some specific cases of coding like in OpenMP, how we can handle the 'nested' clause and nested loops inside this class to accomplish high performance [10]. There is also need to make a strong research on hybrid programming model when we use CUDA, OpenMP, MPI and other models to achieve the high performance computing leading to exascale system.

6. REFERENCES

[1] J. J. Shalf , S. Dosanjh, and J. Morrison, “Exascale Computing Technology Challenges”. Springer-Verlag Berlin Heidelberg. Pp. 1-25. 2011.

[2] J. M. Yusof et al, “Exploring weak scalability for FEM calculations on a GPU-Enhanced cluster”, 33.685–699. Nov, 2007.

[3] C.T. Yang, C.L. Huang and C.F. Lin, “Hybrid CUDA, OpenMP, and MPI parallel programming on multicore GPU”. Computer Physics Communications. Pp. 266-269. 2011.

[4] M. Suß and C. Leopold, “Common Mistakes in OpenMP and How To Avoid Them”. 2007.

[5] J.P. Hoeflinger and B.R. Supinski, “ The OpenMP memory model”. In: Proceedings of the First International Workshop on OpenMP - IWOMP .2005.

[6] D. A. Mey and T. Reichstein, “Parallelization with OpenMP and MPI A Simple Example (Fortran)”. Oct, 2007

[7] M. Zheng, V.T. Ravi, F. Qin, and G. Agrawal , “GRace: A Low-Overhead Mechanism for Detecting Data Races in GPU Programs”. ACM. Dec, 2011.

[8] G. Hager, G. Jost and R. Rabenseifner “Communication Characteristics and Hybrid MPI/OpenMP Parallel Programming on Clusters of Multi-core SMP Nodes”. Cray User Group Proceedings. 2009.

[9] D. Shreiner, M. Woo, J. Neider and T. Davis, “OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL(R)”, Version 2.1, 6th edition, Addison–Wesley Professional, Reading, MA, ISBN 0321481003, 2007.

[10] J. Gustedt, “Parallelizing nested loop in OpenMP,”,http://stackoverflow.com/questions/19193725/ parallelizing-nested-loop-in-openmp-using-pragma-parallel-for-shared, Oct. 5, 2013 [May 10, 2015]

[11] Alrikai, “CUDA racecheck,”http://stackoverflow.com/questions/13861017/cuda-racecheck-shared-memory-array-and-cudadevicesynchronize, Jan. 11,2013 [April 22, 2015]

[12] Daedalus, “How do CUDA blocks/threads map onto CUDA cores,”http://stackoverflow.com/questions/ 10460742/how-do-cuda-blocks-warps-threads-map-onto-cuda-cores, May 5, 2012 [May 14, 2015]

Biclustering using Parallel Fuzzy Approach for Analysis of Microarray Gene Expression Data

Dwitiya Tyagi-Tiwari *dwitiya.sr@gmail.com*
Department of Mathematics & Computer Applications
Maulana Azad National Institute of Technology
Bhopal-462051, India

Sujoy Das *sujdas@gmail.com*
Department of Mathematics & Computer Applications
Maulana Azad National Institute of Technology
Bhopal-462051, India

Manoj Jha *m-jha28@rediffmail.com*
Department of Mathematics & Computer Applications
Maulana Azad National Institute of Technology
Bhopal-462051, India

Namita Srivastava *sri.namita@gmail.com*
Department of Mathematics & Computer Applications
Maulana Azad National Institute of Technology
Bhopal-462051, India

Abstract

Biclusters are required to analyzing gene expression patterns of genes comparing rows in expression profiles and analyzing expression profiles of samples by comparing columns in gene expression matrix. In the process of biclustering we need to cluster genes and samples. The algorithm presented in this paper is based upon the two-way clustering approach in which the genes and samples are clustered using parallel fuzzy C-means clustering using message passing interface, we call it MFCM. MFCM applied for clustering on genes and samples which maximize membership function values of the data set. It is a parallelized rework of a parallel fuzzy two-way clustering algorithm for microarray gene expression data [9], to study the efficiency and parallelization improvement of the algorithm. The algorithm uses gene entropy measure to filter the clustered data to find biclusters. The method is able to get highly correlated biclusters of the gene expression dataset.

We have implemented the algorithm of fuzzy c-means in MATLAB parallel computing platform using MATLABMPI (Message Passing Version of MATLAB). This approach is used to find biclusters of gene expression matrices. The biclustering method is also parallelized to reduce the gene centers with lower entropy filter function. By this function we choose the gene cluster centers with minimum entropy. The algorithm is tested on well-known cell cycle of the budding yeast S. cerevisiae by Cho et al. and Tavazoi et.al data sets, breast cancer subtypes Basal A, Basal B and Leukemia from Golub et al.

Keywords: Biclustering Analysis, Gene Expression, Parallel Computing Toolbox, Fuzzy, MATLABMPI.

1. INTRODUCTION

Large amount of gene expression data demands the need for methods that are effective in analyzing informations that are present in it. One of the primary data analysis tasks is to cluster data which intended to help a user to understand the natural grouping or structure of dataset.

Therefore, the development of improved clustering algorithms have received direct attention. Clustering of genes/samples according to their expression values is an important approach in extracting knowledge from microarray gene expression data [9, 17]. Gene expression data is arranged in a matrix form, where rows are represented as genes and columns are represented the conditions or samples. The main goal of a clustering algorithm is to group the data objects of into a set of meaningful subclasses. A clustering algorithm is often applied on microarray data to find the similar group of genes or samples. These similar expression patterns suggest the co-regulation of the genes or samples. Co-regulation of the genes or samples may possibly be involved in a similar biological functions [7]. Therefore in this paper we have used fuzzy c-means clustering method that can assign single gene/sample to several groups. Further FCM clustering is applied because microarray gene expression data contains noisy components due to biological and experimental factors. Sometimes the gene activity shows large variations under smaller changes of the experimental conditions [9,14]. Clustering algorithms generally follows hierarchical or partitional approaches [15] like k-means, fuzzy c-means algorithm [14], hierarchical clustering etc. However, these clustering approaches used to form clusters in one-way only, i.e. clusters of either gene or sample. Hartigen in 1972 [10] first applied two way clustering or biclustering to describe simultaneous grouping of both row and column subsets in a data matrix. Cheng and Church [19] has used two-way clustering approach for analyze gene expression data. These two-way clustering is called biclusters which are particularly valuable or demanding to mine important information from large databases in both rows and columns dimensions. The literature says that biclustering is NP-hard problem [17], and it takes long time to compute the biclusters, so, it is important to solve the problem in timely manner. In this regard parallel computing technology are the essential solution to minimize execution time of biclustering approach. Parallel approach has also been developed to solve the problem of biclustering in timely manner [2][8][14] Liu and Chen [13] have implemented a parallel algorithm for biclustering of gene expression data is based on anti-monotones property of the quality of data sizes. Parallel identification of gene based biclustering with coherent evolution is identified based on additive modelling by [2].

In this paper, we present fuzzy c-means algorithm for clustering gene and sample dimensions for biclustering using message passing model in MATLAB environment. Message passing interface (MPI) is the real standard for implementing programs on multiple processors in parallel computing. Parallel programs are defines in C, C++ and FORTRAN language functions for doing point to point communication with MPI [1]. We have used MatlabMPI to implement the parallel algorithms. MatlabMPI can run on any combination of computers in which MATLAB can run. Another advantage of using MatlabMPI is that it provides the facility to use MATLAB implicit functions and toolbox [21]. This approach is based upon the previous work of [8]. It is a parallelized rework of a parallel fuzzy two way clustering algorithm for microarray gene expression data, to study the efficiency and parallelization improvement of the algorithm. The outline of this paper is as follows: section 2 gives an overview of parallel biclustering algorithm using fuzzy approach with MatlabMPI; Section 3 describe parallel fuzzy two-way clustering algorithm and discusses parallelization aspects of algorithm; Section 4 presents experimental results; Section 5 discuses conclusion and future work.

2. MATERIALS AND METHODS

2.1 Fuzzy c-means Algorithm

Fuzzy c-means algorithm is originally developed by Bezdek 1981[3], which allows one data point to be present two or more clusters. The algorithm assigns membership values to each data point corresponding to each cluster center on the basis of distance between data point and cluster. Summation of membership values of each data point should be equal to one. The original algorithm is based on minimization of the following objective function:

$$J_m = \sum_{k=1}^{K} \sum_{i=0}^{N} (u_{ki})^m \, \|x_i - c_k\|^2 \qquad \text{.................. (1)}$$

Where *K* is number of cluster and *N* is number of data objects, *m* is constant real-valued number which controls the ‘fuzziness’, u_{ki} is the degree of membership of data object x_i in cluster *k*.

The steps are as follows [3]:

1. **Inputs**: select K i.e. number of cluster and randomly select initial centroids of each cluster.
2. Initialize $U = [u_{ki}]$, $k = 1, \ldots, K$ and $i = 1, \ldots, N$.
   Calculate membership values using formula as given by Bezdek [3]:
   $$u_{ki} = \left\{ \frac{1}{\sum_{s=1}^{K} \left[\frac{\|x_i - c_k\|}{\|x_i - c_s\|} \right]^{\frac{2}{(m-1)}}} \right. \tag{2}$$
3. At l step compute the matrix of centroids $C^{(l)} = c_k$:
   $$c_k = \frac{\sum_{i=1}^{N} (u_{ki})^m x_i}{\sum_{i=1}^{N} (u_{ki})^m} \text{ k=1,2-------K} \tag{3}$$
4. Update $U^l \, to \, U^{l+1}$ by Equation 2.
5. If $U^l = U^{l+1}$ the minimum J is achieved, then terminate; otherwise return to Step 2.

2.2 Message Passing Interface

High performance parallel programming environment uses message passing to communicate and exchanging the data among the processors. MPI is a language independent communications protocol which provide high performance scalability and portability [1][25]. MPICH is a standard for message passing used in parallel computing. MATLAB provides the facility of parallelism through MatlabMPI. MatlabMPI developed at Lincoln Laboratory at Massachusetts Institute of Technology (MIT) designed by Dr. Jeremy Kepner, which is a set of MATLAB scripts or programs that implements a subset of MPI [13]. It allows MATLAB program to be run on a parallel computer. MatlabMPI will run on any combination of computers that MATLAB supports [13][24]. Table1 shows six basic MPI function of MatlabMPI:

TABLE 1: MATLABMPI Uses Six Basic MPI Functions.

MPI Functions	**Function Description**
MPI_Init	To initialize MPI
MPI_Comm_rank	Access Id current processor within a communicator
MPI_Comm_size	Access the number of processors in a communication
MPI_Send	Sends the data or message to processor
MPI_Recv	Receives the data or message from a processor
MPI_Finalize	To terminate or Finalizes MPI

All of the MPI programs establish the communication between the concurrent execution parts. It initializes and starts with MPI-Init(), then establish the processes and new communicator functions and applications to be used for each process, and in last it terminates with MPI_Finalize(). The flow of parallel program design with message passing is shown in figure 1 [25].

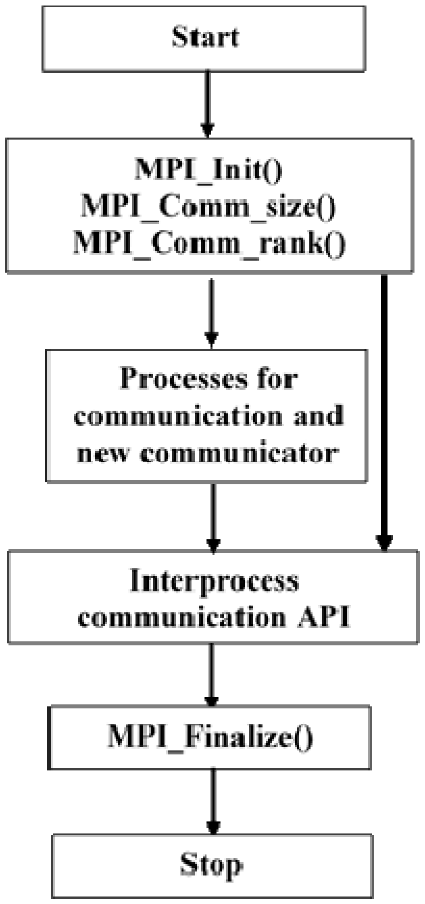
Start
MPI_Init()
MPI_Comm_size()
MPI_Comm_rank()
Processes for communication and new communicator
Interprocess communication API
MPI_Finalize()
Stop

FIGURE 1: Message Passing Interface Process Flow.

MatlabMPI is also used basic MPI function to write a script on MATLAB these are defined in Table1. The basic steps for executing MatlabMPI is given in [13][24]. These steps are included as follows [13]:

1. To set the path for MatlabMPI source in MATLAB current directory.
2. Initialize MPI library.
3. Create the communicator.
4. Define the size and rank of the communicator.
5. Computation using MATLAB.
6. Communication between the processes using MPI for exchanging data.
7. Terminate MPI application.

2.3 Proposed Parallel Fuzzy c-means Algorithm

The fuzzy clustering method assigns one gene/sample to multiple clusters according to their membership values. So that this method is more appropriate for analyzing gene expression profiles of gene expression data, because a single gene might be involved in multiple functions. Main objective of this algorithm is to improve the performance of fuzzy c-means approach for biclustering of gene expression data by distributing computation and main memory usage. The algorithm is based on data parallelism and communicated between the processes using MatlabMPI [13] with SPMD (Single Program Multiple Data) parallel computing model for data distribution.

Based on above original algorithm we have implemented parallel version of FCM using MATLABMPI, as shown in figure 1. The parallel algorithm divides the *N* data points in the *M* processors. Here master processor is MATLAB client and MATLAB processors are participating processors. The proposed parallel fuzzy c-means algorithm is as follows:

```
Input: number of clusters K, number of data matrix N.
Output: K centroids. Initialize random cluster centroids;
        Distribute the N data matrix among all processors in N/P.

    1.  MPI_Init();  %start the procedure
    2.  M = MPI_Comm_size();
    3.  Id = MPI_Comm_rank();
    4.  Initialize the random cluster centroid for each data point in fuzzy cluster
        K;
    5.  Max-itr = 0;
    6.  while ((Max_itr == 100) || (Old_ Mem_fcn[j i] == Mem_fcn[i j]))
    7.  For i = 1 : K
    8.  Mat_of_centr[i j] = 0;  % stores matrix of centroids value Eq2.
    9.  Val_centr[i j] = 0;  %  stores the value of cluster centroid
    10. Mem_fcn[i j] = 0;  %  membership function for next iteration Eq3
    11. end;   %  end for
    12. for i = id * (N/M) + 1 : (Id + 1) * (N/M)
    13. for j = 1 : K
    14. Update Mat_of_centr[K];
    15. Update Val_centr[K];
    16. end;   % End for j
    17. end;     %  End for i
    18. for j = 1 : K
    19. MPI_reduce(Mat_of_centr[j], Mat_centrP [j], MPI_SUM); %
        Mat_centrP- local matrix of centroids.
    20. MPI_reduce(Val_centr[j], Val_centrP[j], MPI_SUM); %
        Val_centrP- local values of centroids.
    21. Update centroid vectors:
        Val_centrP[j] = Val_centrP[j] / Mat_centrP [j];
    22. end; %  End for j
    23. for i = id * (N/M) + 1 : (Id + 1) * (N/M)
    24. for j = 1 : K
    25. Update Mem_fcn[j i];
    26. Old_ Mem_fcn[j i] = Mem_fcn[i j];
    27. Max_itr = Max_itr + 1;
    28. end; %  End for i
    29. end;  % End for j
    30. MPI_reduce(max_itr, itr, MPI_MAX);
    31. end;  % End While.
    32. MPI_Finalize();
```

FIGURE 2: Algorithm1- Fuzzy C-means clustering procedure with MATLABMPI.

The algorithm is modified by implementing Fuzzy C-means at Step1 and 2 of Tang et al. [5] and has further implemented on MATLAB using Message Passing Interface. Gene reduction is done using gene entropy instead of correlation coefficient.

Main algorithm steps for parallel interrelated two-way clustering procedure are as follows:

1. Genes clustering: Parallel Fuzzy C-means algorithm is given in the previous section is used to cluster the genes using algorithm1 of figure 2.

2. Sample clustering: Parallel Fuzzy C-means algorithm is given in the previous section is used to cluster the samples algorithm1 of Figure 2.
   In Step 1 and 2 data points are divided into *K* groups which take the number of clusters as an input parameter. Here we take *K* = 4 and 6.

3. Find gene centers: Maximize the sum of distances between the genes having maximum membership value from membership matrix.

4. Gene centers reduction by gene entropy filter: Apply low gene entropy filter function on gene cluster centers found in Step2 and remove two gene cluster centers and reduce the genes of same cluster centers.

   This process is also parallelized to give one cluster to each processor to calculate gene entropy and give result back to master processor. Clients find the lower gene entropy and remove.

   Entropy is a function of correlation which provides the amount of information that may be gained by an observation of a system and it measures variation or changes in a series of events. Entropy denotes the diversity of information in given data set which makes it suitable for clustering genes. We calculate the low gene entropy of given gene expression datasets. For the measurement of interdependency of two random genes X and Y we have used a direct MATLAB function:

   [Mask, FData, FNames] = geneentropyfilter(GED-MATRIX, Names, 'Percentile', PercentileValue)

5. Termination condition: Repeat the process and compute the gene and sample groups with Step1. Stop the iteration if the iteration reached 50; otherwise continue. If the number of genes dropped is less than 15% of the total number of genes, the iterations are stopped.

Figure 2 shows the block diagram of parallel fuzzy c-means approach. The algorithm follows the master/slave modelling approach for implementation. Here master processor have MATLAB and others are the slaves in which MATLAB workers runs. All the processors can share the messages passing between them.

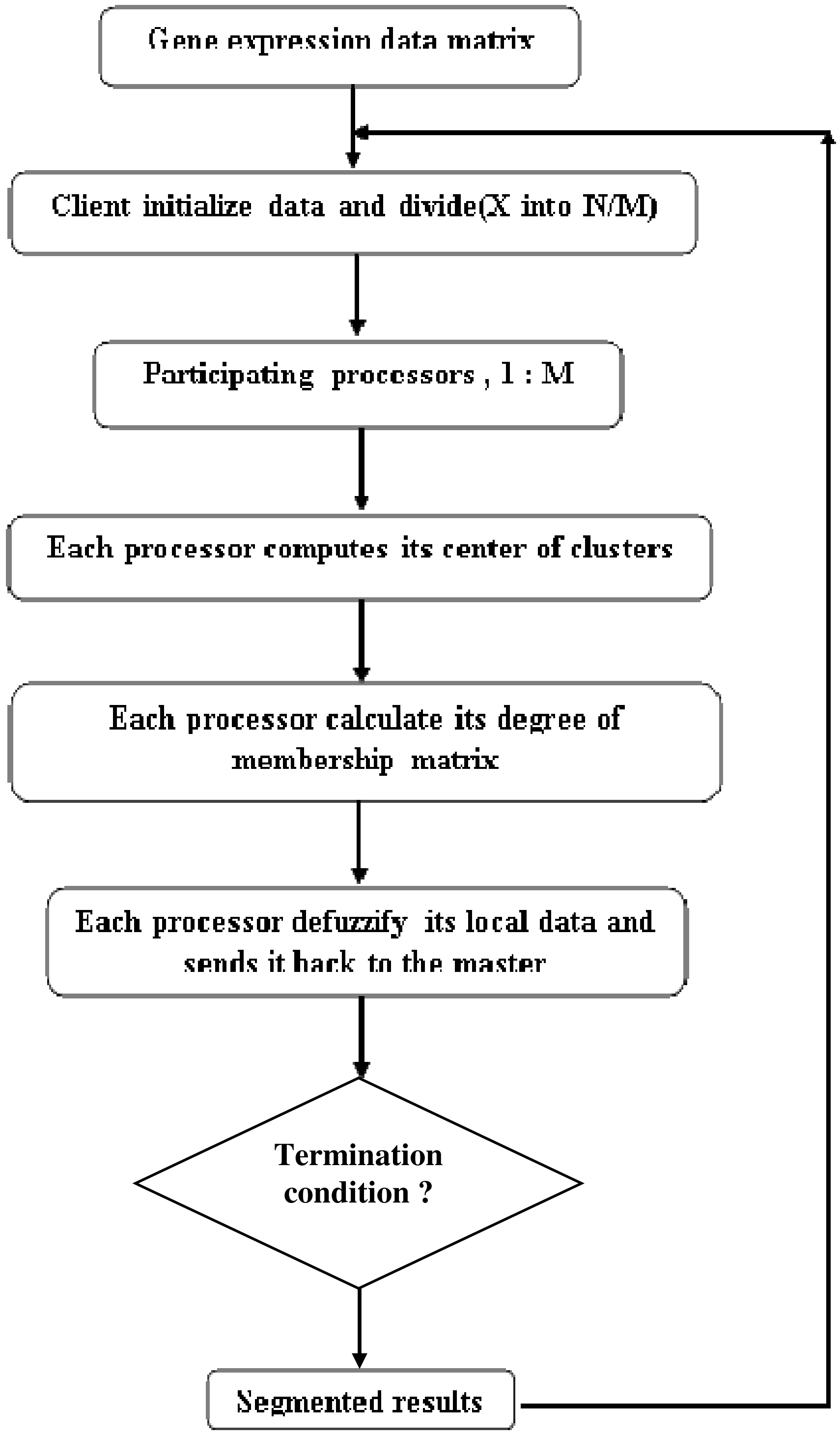
Gene expression data matrix
Client initialize data and divide(X into N/M)
Participating processors , 1 : M
Each processor computes its center of clusters
Each processor calculate its degree of membership matrix
Each processor defuzzify its local data and sends it back to the master
Termination condition ?
Segmented results

FIGURE 3: Block Diagram of Parallel Fuzzy c-means Approach.

3. RESULT & DISCUSSION

Microarray Yeast cell cycle data: In this paper we have used cell cycle of the budding yeast S. cerevisiae made by Cho et al. The data set contains 6149 genes were measured every 10 minutes under 17 time points [6]. Other yeast cell cycle data which contains the expression profiles of 6200 yeast genes were measured every 10 minutes during two cell cycles in 17 hybridization experiments [20]. In the selection of the data set for the time points 90, 100 and 120 minutes are excluded. Simple description of data sets is shown in Table1.

We have applied this algorithm on other two subtypes of breast cancer data sets named as subtype Basal A and Basal B [21].

TABLE 2: Description of Dataset Used in Study

Data Set	Dataset code	Data Size (kb)	No. of genes	No. of samples
yeast	Data1	2309	6149	17
yeast	Data2	939	2945	15
Basal A	Data3	776	1213	98
Basal B	Data4	521	1213	49
Leukemia	Data5	7222	12560	72

3.1 Performance Evaluation

We examined parallel programs for six to eight processors. The results are shown here is based on six processor computation to find the biclusters from the data sets. We have calculated 30 biclusters for each dataset and the execution time are shown in Table3, 4, 5 and 6 and graphs 4 and 5. The main objective of this study is to reduce the processing time to find the biclusters of the data. The CPU time measured of CPU time for clustering of genes and samples and is shown in Table3. A general rule of thumb is that a clustering result with lower CPU time is preferable. For a comparable assessment, we coded these methods by using the fuzzy tools available in MATLAB with MatlabMPI for parallelization [18, 19, 22]. Here we use the SPMD (Single Program Multiple Data) parallel computing model to parallelize the algorithm [20]. In SPMD the dataset is divided into number or processors and a single program will work on all the processor's local datasets.

Table4 reports the results of computation time taken by the algorithms, including clustering time of genes and samples, bicluster computation time and the total running time. Table4 shows compare that running time cost of the serial biclustering approach is higher than the parallel biclustering including parallel fuzzy c-means with MPI.

Table5 illustrates the total time overheads of serial fuzzy c-means of genes and samples, and applying MFCM on genes and samples respectively. Figure 4 and 5 show total computation time of Step 1 and 2 of the proposed biclustering algorithm when number of processors are two. We see that the execution time to cluster genes and sample in case of serial fuzzy clustering is higher than the MPI based fuzzy clustering.

TABLE 3: The Performance Result of Serial and Parallel Fuzzy c-means For Gene Based Clustering with Four Processors and K = 4.

Dataset code	FCM (sec)	MFCM (sec)
Data1	29.3829	12.9292
Data2	12.9276	10.0113
Data3	17.1909	9.0931
Data4	17.1206	10.926
Data5	48.7918	31.7711

TABLE 4: Performance Evaluation of the Complete Execution time of Serial and Parallel Biclustering with Two Processors and K=6.

Data set	Biclustering using FCM (sec)			Biclustering using MFCM (sec)		
	Clustering	Biclustering	Total	Clustering	Biclustering	Total
Data1	31.289	39.12	70.409	15.0192	19.014	34.0332
Data2	18.901	28.5	47.401	10.9018	11.0112	21.913
Data3	19.109	21.082	40.191	10.0991	11.7919	21.891
Data4	19.208	20.774	39.982	11.828	8.94	20.768
Data5	52.998	47.203	100.201	32.6431	18.1779	50.821

The parallel execution time is the elapsed time from when parallel computation starts to the moment when the last processor finishes its computation [1]. Here the sequential execution time is denoted by *TSeq* and the parallel execution time is denoted by *TPar*. Speedup calculation is defined as the ratio of the sequential execution time and the parallel execution time to solve the same problem in sequential and parallel computers respectively.

The speedup is calculated according to the following equation:

$$Speedup = \frac{TSeq}{TPar} \qquad (6)$$

Table5 shows the execution time difference between serial and parallel algorithm, and speedup performance of the parallel biclustering algorithm calculated for k = 6 clusters of genes/samples for Step1 and 2. The parallel calculated time is of biclustering algorithm is based on four processors.

TABLE 5: Speedup Performance for the Biclustering of the Dataset with Time Difference Between Serial and Parallel Approach on Six Processors and Six Clusters.

Dataset Code	Number of Clusters	TSeq	TPar	TDiff	Speedup
Data1	6	70.409	22.121	48.288	3.183
Data2	6	47.401	12.01	35.391	3.947
Data3	6	40.191	12.989	27.202	3.094
Data4	6	39.982	11.5249	28.4571	3.4692
Data5	6	100.201	43.9981	56.2029	2.277

Efficiency is a measure of the fraction of time for which a processor is usefully employed. It is defined as the ratio of speedup to the number of processors (P). We examined parallel programs for six to eight processors. Efficiency by the symbol *Eff*:

$$Eff = \frac{Speedup}{P} \qquad (7)$$

TABLE 6: Efficiency Calculations.

Dataset	Number of processors	TPar	Speedup	Efficiency
Data1	2	34.0332	2.068832787	1.034416393
	4	30.119	2.337693815	0.584423454
	6	22.121	3.182903124	0.530483854
	8	19.225	3.66236671	0.457795839
	2	21.913	2.163145165	1.081572582

Data2	4	19.2213	2.466066291	0.616516573
	6	12.01	3.946794338	0.657799056
	8	10.09	4.697819623	0.587227453
Data3	2	21.891	1.835959984	0.917979992
	4	20.212	1.988472195	0.497118049
	6	12.989	3.094233582	0.515705597
	8	11.2201	3.582053636	0.447756704
Data4	2	20.768	1.925173344	0.962586672
	4	20.001	1.99900005	0.499750012
	6	11.5249	3.469184114	0.578197352
	8	10.771	3.712004456	0.464000557
Data5	2	50.821	1.97164558	0.98582279
	4	45.662	2.194406728	0.548601682
	6	43.9981	2.277393797	0.379565633
	8	42.2332	2.372564712	0.296570589

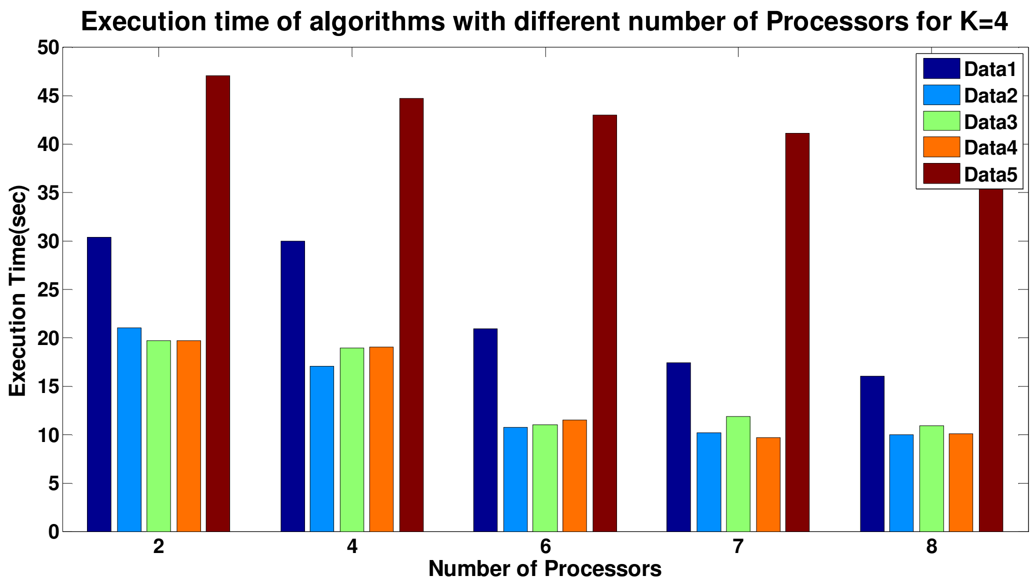

FIGURE 4: Execution time with proposed method of Datasets 1,2,3,4 and 5 with number of processors for K=6.

Figure 4 shows that the running time of biclustering with serial fuzzy c-means and biclustering with parallel fuzzy c-means with MATLABMPI when number of clusters of genes and samples is 4. It is clear from figure 4 that execution time is reducing when we increase the number of processors.

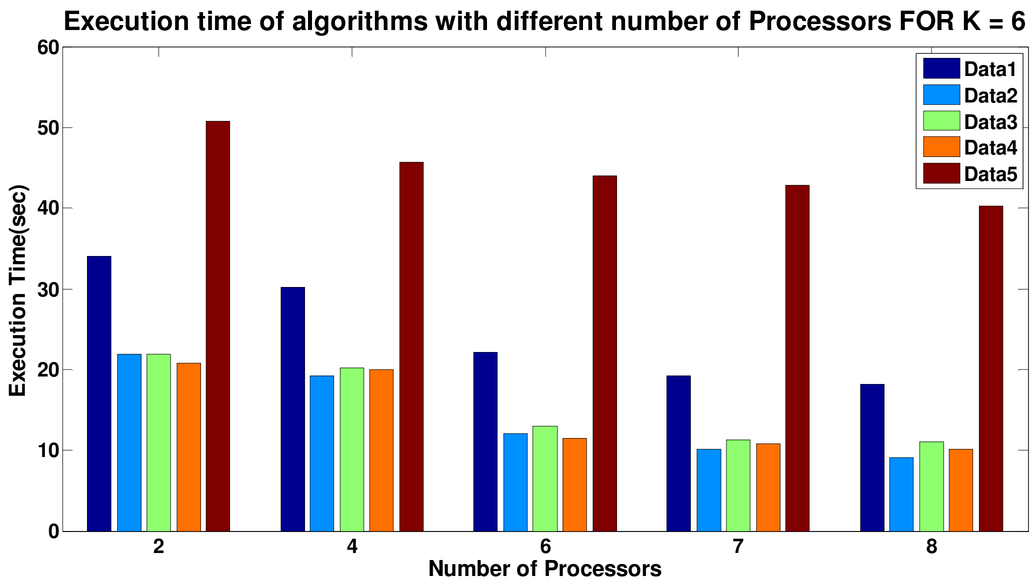

FIGURE 5: Execution time with proposed method of Datasets 1,2,3,4 and 5 with number of processors for K=6.

Figure 5 shows that the running time of biclustering with serial fuzzy c-means and biclustering with parallel fuzzy c-means with MatlabMPI when the number of clusters of genes and samples is 6. From the figure 5 it is clear that execution time is going lower when we increase the number of processors. After sixth processor the graph is showing little bit difference between execution time of the algorithm for all the data sets. The formation of biclusters also indicates that a gene or a sample may belong to more than one cluster, which in turn may be helpful in understanding the nature of biological process.

4. CONCLUSION

In this paper, we have presented parallel biclustering algorithm for yeast, cancer subtypes (BasalA, BasalB), and Leukemia microarray gene expression data used in previous study by Cho et al. and Tavazoie et al. We proposed a parallel biclustering algorithm based on MATLABMPI (MFCM) to find the genes and samples clusters. It is clear that serial algorithm needs more time to compute the clusters from datasets, and the biclustering also takes higher execution time. Parallel technology provides efficient tools to let these data clusters execute on a computer cluster. The algorithm is based on data parallel computing model, and is implemented with MatlabMPI. Experimental results show the feasibility and effectiveness of parallel algorithms of clustering and biclustering of the gene expression data. Message passing is the communication between the processes, then some time it is also possible that it takes longer time to communicate with other processes for large datasets, but it gives better performance than previously implemented algorithm [9].

This paper presented algorithms for finding biclusters from gene expression data in parallel, further exploration on the resulting biclusters will be another interesting research approach to reveal gene functions, gene regulations, and cellular process of gene regulatory networks.

5. ACKNOWLEDGEMENT

One of the authors Dwitiya Tyagi-Tiwari would like to thank Åbo Akademi ComBio Lab to provide me space and resources for working and staff for helpful discussion.

6. REFERENCES

[1] Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, “Introduction to parallel computing”, Addison-Wesley, 2003.

[2] A.H. Tewfik, A.B. Techagang and I. Vertatsehitsch, “Parallel Identification of Gene Biclusters with Coherent Evolutions”, IEEE Transaction on Signal Processing, Vol. 54, No. 6, June-2006.

[3] Bezdek,J.C., “Pattern Recognition With Fuzzy Objective Function Algorithms”, Plenum Press, New York, 1981.

[4] B. Chandra, S. Shankera, Saroj Mishra, "A new approach: Interrelated two-way clustering of gene expression data", Statistical Methodology 3, 2006, pp. 93–102.

[5] Chun Tang and Aidong Zhang, “Interrelated Two-Way Clustering and Its Application on Gene Expression Data ", International Journal on Artificial Intelligence Tools, 2005; Vol. 14, No. 4; pp. 577-598.

[6] Cho RJ, Campbell MJ, Winzeler EA, Steinmetz L, Conway A, Wodicka L, Wolfsberg TG, Gabrielian AE, Landsman D, Lockhart DJ, Davis RW, ‘A genome-wide transcriptional analysis of the mitotic cell cycle’, Molecular Cell, Vol. 2, 65–73, July, 1998.

[7] Dembele D, Kastner P. Fuzzy C-means method for clustering microarray data. Bioinformatics 2003; 19(8):973–80.

[8] Dwitiya Tyagi-Tiwari, Sujoy Das, and Namita Srivastava, Parallel Two-way Clustering for Microarray Gene expression data’, International Journal of Computer Science Trends and Technology, Vol. 3 Issue 3, May-June 2015.

[9] Dwitiya Tyagi-Tiwari, Sujoy Das and Namita Srivastava. Article: Two-Way Clustering Analysis using Parallel Fuzzy Approach for Microarray Gene Expression Data. International Journal of Computer Applications 124(9), pp. 39-45, August 2015.

[10] G. Kerr, H.J. Ruskin, M. Crane and P. Doolan, “Techniques for clustering gene expression data”, Computers in Biology and Medicine 38, pp. 283-293, 2008.

[11] Hartigan J.: “Direct Clustering of a Data Matrix”, J Am Stat Assoc 1972, 67(337), pp. 123-129.

[12] Huimin Geng, Dhundy Bastola, and Hesham Ali, “A New Approach to Clustering Biological Data Using Message Passing”, Proceedings of the IEEE Computational Systems Bioinformatics Conference, 2004.

[13] Jeremy Kepner, “Parallel programming with MATLABMPI”, 2002, High Performance Embedded Computing (HPEC) workshop, MIT Lincoln Laboratory, Lexington, MA, http://arXiv.org/abs/astro-ph/0107406

[14] Liu Weihj And Chen Ling, “*A Parallel Algorithm for Gene Expressing Data Biclustering*”, Journal Of Computers, Vol. 3, No. 10, October 2008.

[15] Li Li, Yang Guo, Wenwu Wu, Youyi Shi, Jian Cheng and Shiheng Tao, 'A comparison and evaluation of five biclustering algorithms by quantifying goodness of biclusters for gene expression data', BioData Mining 2012, Vol-8 pp. 1756-0381.

[16] Matthias E. Futschik and Nikola K. Kasabov, “Fuzzy Clustering of Gene Expression Data”, 2002 IEEE International Conference on Fuzzy Systems, 2002, Vol 1, pp. 414-419.

[17] MATLAB the MathWorksTM Accelerating the pace of engineering and science Parallel Computing Toolbox™ 4 User's Guide, 2009.

[18] Sara C. Madeira and Arlindo L. Oliveira, "Biclustering Algorithms for Biological Data Analysis: A Survey", IEEE/Acm Transactions on Computational Biology and Bioinformatics Vol 1, No. 1, January-March 2004, pp. 24-45.

[19] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gassenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing,M.A. Caligiuri, D.D. Bloomfield, E.S. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (15) (1999) pp 531–537.

[20] Terence Kwok, Kate Smith, Sebastian Lozano and David Taniar, "Parallel Fuzzy c-Means Clustering for Large Data Sets", Springer-Verlag Berlin Heidelberg 2002, LNCS 2400, pp. 365-374.

[21] Yizong Cheng and George M. Church, "*Biclustering of Expression Data*", Proc. ISMB'00, pp. 93-103, 2000.

[22] Tavazoie,S., Hughes,J.D., Campbell,M.J., Cho,R.J. and Church,G.M. (1999) Systematic determination of genetic network architecture. Nat. Genet., 22, 281–285.

[23] Hoshida Y, Brunet J-P, Tamayo P, Golub TR, Mesirov JP Subclass Mapping: Identifying Common Subtypes in Independent Disease Data Sets. PLoS ONE 2(11), 2007.

[24] https://www.ll.mit.edu/mission/cybersec/softwaretools/MATLABmpi/MATLABmpi.html.

[25] https://www.mpich.org/documentation/guides/.

Managing Intrusion Detection Alerts Using Support Vector Machines

Majid Ghonji Feshki *ghonji.majid@yahoo.com*
Department of computer science
Islamic Azad University, Qazvin Branch
Qazvin, Qazvin, Iran

Omid Sojoodi Shijani *o_sojoodi@qiau.ac.ir*
Department of computer science
Islamic Azad University, Qazvin Branch
Qazvin, Qazvin, Iran

Minoo Deljavan Anvary *Minoo.deljavan@yahoo.com*
IT Department School of e-Learning
Shiraz University
Shiraz, Fars, Iran

Abstract

In the computer network world Intrusion detection systems (IDS) are used to identify attacks against computer systems. They produce security alerts when an attack is done by an intruder. Since IDSs generate high amount of security alerts, analyzing them are time consuming and error prone. To solve this problem IDS alert management techniques are introduced. They manage generated alerts and handle true positive and false positive alerts. In this paper a new alert management system is presented. It uses support vector machine (SVM) as a core component of the system that classify generated alerts. The proposed algorithm achieves high accurate result in false positives reduction and identifying type of true positives. Because of low classification time per each alert, the system also could be used in active alert management systems.

Keywords: Intrusion Detection System, Alert Management, Support Vector Machine, Security Alert Classification, Reduction of False Positive Alerts, Classifying True Positive Alert Based on Their Attack types.

1. INTRODUCTION

An intrusion detection system (IDS) monitors network traffic and suspicious activity and then alerts the system or network administrator. In some cases the IDS may also responds to anomalous or malicious traffic by taking action such as blocking the user or source IP address from accessing the network [1]. Common problem with IDSs is producing many alerts each day that many of them are false positive alerts. Since amount of fake alerts is high so alerts of real attacks are hided among them. IDSs divided in to two categories, passive and active. In passive usage of IDS, it analyzes traffics or events in offline mode but active IDSs work in online mode. To manage alerts concurrently with alerts generation, active alert management systems are used. Active alert management systems same as active IDSs, work in online mode. Some of problems of IDS are: huge amount of generated alerts and high rate of false positive alerts among generated alerts. Also most alert management system has low speed, and has low accuracy in classification results. In this paper a new alert management system proposed that uses Support Vector Machine (SVM) as a classification engine [2]. It classifies the generated alerts based on attack type of alerts, detects false positive alerts, high speed classification to use with alert generation in IDSs. The proposed system uses some techniques such as alert filtering, alert preprocessing, and alert filtering to improve accuracy of the results.

In Section 1 the alert management system is introduced. Section 2 reviews related works, section 3 explains the suggested alert management system and describes all component of the proposed system, the experimental results are shown in section 4 and finally section 5 is a conclusion and future works.

2. RELATED WORKS

Clustering and classification of alerts is a technique of alert management systems. A method of clustering based on root causes is proposed by K. Julisch [3] which clusters IDS alerts by discovering main cause of their occurrences. He proves that a small number of root causes imply 90% of alerts. By removing alerts related with these root causes total number of alerts come down to 82%. The system uses information about underlying network so it is not portable and this problem is a disadvantage of the technique. In [4, 5] two genetic clustering algorithm based, named Genetic Algorithm (GA) and Immune based Genetic Algorithm used to manage IDS alerts. Their proposed methods are depended on underlying network information same as method presented by Julisch.

Three algorithms with dimension reduction techniques are used to cluster generated IDS alerts from DARPA 2000 dataset [6]in [4] and then compared results. The problems of that system are: row alert without preprocessing are entered to the algorithms and system is not tuned. Cuppens proposed another method as a part of MIRADOR project that uses expert system to make decision [7, 8].

Debar et al. [9] designed a system by placing them in situations aggregates alerts together. Situations are set of special alerts. Some attributes of alert are used to construct a situation.

In [10] Azimi et.al. is proposed a new system that manage alerts generated from DARPA 98 dataset with snort. Some algorithm such as alert filtering, alert preprocessing and cluster merging are used in that system. The main unit of the system is cluster/classify unit that uses Self-Organizing Maps (SOM) [11] to cluster and classify IDS alerts. Results of [10] show that SOM was able to cluster and classify true positive and false positive alerts more accurate than other techniques.

In another work, authors of previous article develop an alert management system [12] similar to [10]. In that work usage of seven genetic clustering algorithms named Genetic Algorithm (GA) [13], Genetic K-means Algorithm (GKA) [14], Improved Genetic Algorithm (IGA) [15], Fast Genetic K-means Algorithm (FGKA) [16], Genetic Fuzzy C-means Algorithm (GFCMA) [17], Genetic Possibilistic C-Means Algorithm (GPCMA) [12] and Genetic Fuzzy Possibilistic C-Means Algorithm (GFPCMA) [12] to cluster and classify true positive and false positive alerts, are explained. The system after clustering alerts then prioritized produced clusters with Fuzzy Inference System [12].

Also they were develop another approach based on same alert management system in [10, 12] by replacing classification engine of previous work by Learning Vector Quantization (LVQ) [18, 19]. In that work LVQ technique is used in the classification unit. The experimental results shows that LVQ has can be a solution to alert management systems that used in active mode because of high classification speed. The accuracy of the suggested approach is acceptable [18].

In [20] a fast and efficient vulnerability based approach that addresses the above issues is proposed. It combines several known techniques in a comprehensive alert management framework in order to offer a novel solution. Their approach is effective and yields superior results in terms of improving the quality of alerts.

Alert Correlation is one of alert management techniques is used to find any relation between alerts and tries to find attack scenarios. In a new alert correlation technique named ONTIDS is proposed. It is a context-aware and ontology-based alert correlation framework that uses ontologies to represent and store the alerts information, alerts context, vulnerability information, and the attack scenarios. ONTIDS employs simple ontology logic rules written in Semantic Query-enhance Web Rule Language (SQWRL) to correlate and filter out non-relevant alerts. they illustrate the potential usefulness and the flexibility of ONTIDS by employing its reference implementation on two separate case studies, inspired from the DARPA 2000 [21] evaluation datasets.

In this paper an alert management system based on system proposed by Azimi et.al. [10] is proposed that uses SVM as a tool to classify input alert vectors. It covers the problems of previous work such as low accuracy of results, inability to identify the types of generated alerts accurately. The system will be able to improve accuracy of results, to detect type of true positive alerts and also to reduce the number of false positive alerts.

3. USING SVM IN ALERT MANAMENT SYSTEM

In this section we investigate alert management framework introduced by [10, 12]. The suggested framework consists of several units. In this investigation we use Snort [11] IDS to generate alerts from DARPA 98 dataset [22]. Figure 1 shows the proposed framework. Snort is an open source, network based and signature based IDS. Snort gets tcpdump binary files of DARPA 98 dataset as input and produces security alerts. After producing alerts with snort; they are entered to labeling unit. The labeling unit and others units are investigated in next sections.

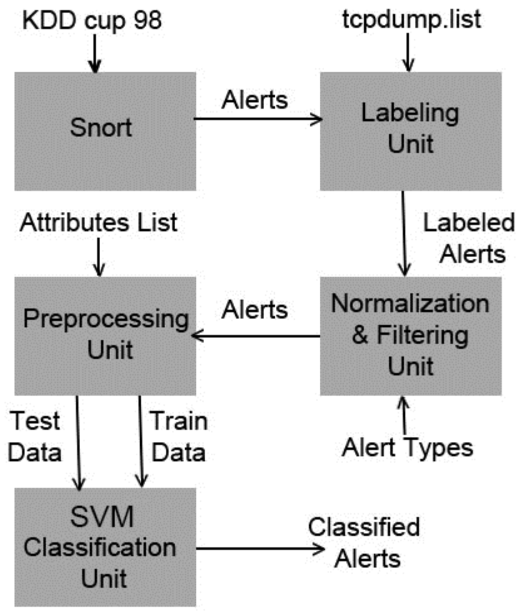

FIGURE 1: Proposed alert management system.

3.1 Labeling Unit

Labeling unit according to tcpdum.list files label generated security alerts. It means that this unit appends attack type of each alert as an attribute of processed alert to proper alert. This operation used in next unit to train and to specify correctness of generated results of the system. Figure 2 shows the labeling algorithm.

```
1. Input TCPDUMP list files.
2. Input alert log files.
3. Create an empty AttackList set.
4. Create an empty AlertList set.
5. For each row in TCPDUMP list files:
   5.1. If the row is a labeled attack then add
        the row to the AttackList set.
6. For each row in alert log files:
   6.1. Create key with the five attributes:
        source IP, destination IP, source port,
        destination port, ICMP code/type.
   6.2. If the key exists in AttackList set then
        label the selected row with the type of
        found attack from AttackList set.
       Else
       Label the selected row with the False
       Positive attack type.
   6.3. Add the selected row to the AlertList
        set.
7. Return the AlertList set.
```

FIGURE 2: Alert labeling algorithm used in Labeling Unit [10, 12].

3.2 Normalization and Filtering Unit

According to [23] snort is unable to detect all attacks in DARPA 98 dataset then we use “alert types” file to specify accepted attack types in this investigation [10, 12]. In this unit alerts according to its labels are filtered. All of accepted attack types are entered to this unit and then this unit removes all of alerts that their labels are not in “alert types” file. Also this unit removes redundant alerts from set and keeps one of them. In this paper accepted alerts attack types are: Back, Land, Pod, Phf, Rootkit, Nmap, Imap, and Dict.

3.3 Preprocessing Unit

This unit transforms character form values of attributes of alert to numerical data to construct data vectors from alerts (1) and (2). The range of attribute values is reduced by this unit (3).

$$IP = X_1.X_2.X_3.X_4, \qquad (1)$$
$$IP\_VAL = (((X_1 \times 255) + X_2) \times 255 + X_3) \times 255 + X_4$$

$$protocol\_val = \begin{cases} 0, protocol = None \\ 4, protocol = ICMP \\ 10, protocol = TCP \\ 17, protocol = UDP \end{cases} \qquad (2)$$

$$IUR = 0.8 \times \frac{x - x_{\min}}{x_{\max} - x_{\min}} + 0.1 \qquad (3)$$

3.4 SVM Training and Classification Unit

In this unit SVM algorithm is used as a classifier. SVM is a technique to solve classification problems [2]. It maps pattern vectors to a high dimensional feature space to generate best separating hyperplane. SVM uses linear model to implement nonlinear models. When a linear model constructed in new space can represent nonlinear decision boundaries in original space. One of main characteristics of SVM is that it simultaneously minimizes the empirical classification error and maximizes the geometric margin [24]. There are many types of kernels that may be used in an SVM [25]. Some nonlinear kernel type is Polynomial and Gaussian (radial based Kernel). Equations Polynomial and Gaussian are respectively.

$$\left(coeff + x^T y\right)^d \qquad (4)$$

$$\exp\left(-|x-y|^2 / \delta^2\right) \qquad (5)$$

In this paper we use Polynomial and Gaussian kernel functions to evaluate SVM classifier.

4. EXPERIMENTAL RESULTS

To simulate the proposed system C#.net programming language, MATLAB and OSU SVM toolbox are used[26, 27]. The parameters of simulation are shown below.

In the investigation three type of kernel used for SVM named Linear, Polynomial and Radial Basis Function (RBF) kernels. Each of these kernels has own parameters for example Linear SVM depends on C and Epsilon parameters where Epsilon is used to termination threshold. The C parameter tells the SVM optimization how much you want to avoid misclassifying each training example. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points. For very tiny values of C, you should get misclassified examples, often even if your training data is linearly separable. Polynomial kernel depends of four parameters named Degree, Gamma, Coefficient and C. Also RBF kernel depends on two factor named Gama and C.

The attack types used in this simulation are: Back, Pod, Nmap, Imanp, Dict, Rootkit, Land and Phf. Train data contains 70% of total filtered alert data vectors or 10166 data vectors. The false positive count in the training dataset is 4113. Test dataset includes 30% of the data vectors of labeled alerts; it means 2591 data vectors of true positive, and 1764 data vectors of false positive alerts.

To evaluate the performance of algorithms four measurements are introduced, they are:

1- Classification Error (ClaE),
2- Classification Accuracy percent (ClaAR),
3- Average Alert Classification Time (AACT),
4- False Positive Reduction Rate (FPRR).

According to tables 1 to 3 the classification error rate reduces by increasing the value of C. Results show that RBF kernel is best to classify IDS alerts. According to table 3 when Gama is 10 and C is 16 best results is achieved.

The best values of ClaAR are 99.82%. The value of AACT measurement is 0.0.00032 that shows the proposed system can be used in active IDS alert management systems that evaluate alerts beside alert production by IDS concurrently.

The tables show the results of accuracy of proposed system based on Linear, Non-linear and RBF kernels in identifying attack type of each alert vector in test phase. As it can be seen in table 3, the proposed system can identify all of attack types of alerts with high rate of accuracy when RBF kernel is used.

An important point is accuracy percent of false positive identification. That is the proposed system can reduce false positive alerts with 99.94 percent. Which shows to be a solution of an important problem of IDSs. Proposed alert management system reaches 100 percent for Back, Land, Dict and Nmap attack types. For attack types Phf, Imap and Rootkit accuracy percent values are 66.67, 66.67 and 42.86 respectively. The number of generated SVM is 211.

Bahrbegi et. al. in [12] proposed a framework that uses genetic algorithm families to clustering and classification propose. As two works are similar we have to compare our results with their work. These results are shown in table 3. For all metrics the proposed system has high value in contrast of all GA based techniques. As shown in table 4, these algorithms could not be able to work actively because of the execution times are high. Although the proposed method earns high accuracy results per alert attack type.

C	CAR	Time	SVM No	Back	Land	Pod	Phf	Rootkit	Imap	Dict	NMap	FPRR
1	99.01	0.000065	1029	99.22	0	97.96	0	0	0	100	100	99.15

TABLE 1: Extracted performance metric values from simulation with Linear Kernel.

C	CAR	Time	degree	SVM No	Back	Land	Pod	Phf	Rootkit	Imap	Dict	NMap	FPRR
1	99.01	0.000222	1	1029	99.22	0	97.96	0	0	0	100	100	99.16
1	99.61	0.000082	5	337	99.92	0	97.96	33.33	0	66.67	100	100	99.94
1	99.77	0.000054	10	238	100	100	97.96	66.67	14.29	66.67	100	100	99.94
0.5	99.74	0.00054	10	209	100	100	97.96	66.67	28.57	33.33	100	100	99.89
1	99.77	0.000047	15	209	100	100	97.96	66.67	28.57	66.67	100	100	99.89
1	99.74	0.000039	20	140	100	100	100	33.34	28.57	66.67	100	100	99.83

TABLE 2: Extracted performance metric values from simulation with Non-Linear Kernel.

C	CAR	Time	Gama	SVM No	Back	Land	Pod	Phf	Rootkit	Imap	Dict	NMap	FPRR
1	99.01	0.00017	0.5	972	99.22	0	97.96	0	0	0	100	100	99.15
1	99.33	0.000079	2	511	99.45	0	97.96	0	0	0	100	100	99.77
1	99.31	0.000122	3	443	99.45	0	97.96	0	0	0	100	100	99.72
1	99.63	0.000054	10	326	99.84	100	97.96	0	0	66.67	100	100	99.89
2	99.72	0.000061	10	269	99.92	100	97.96	33.34	42.86	66.67	100	100	99.83
4	99.79	0.000043	10	246	100	100	97.96	33.34	42.86	66.67	100	100	99.94
8	99.79	0.000032	10	224	100	100	97.96	33.34	42.86	66.67	100	100	99.94
16	99.82	0.000032	10	211	100	100	97.96	66.67	42.86	66.67	100	100	99.94
1	99.60	0.000061	15	369	99.84	100	97.96	0	0	66.67	100	100	99.83

TABLE 3: Extracted performance metric values from simulation with Non-Linear Kernel.

Algorithm	ClaE	ClaAR	FPRR	AACT
GA	1218	72.03	52.15	Offline
GKA	1011	75.2	62.11	Offline
IGA	306	92.97	95.24	Offline
FGKA	314	92.79	97.51	Offline
GFCMA	148	96.60	97.51	Offline
GPCMA	91	97.91	96.03	Offline
GFPCMA	148	96.60	97.51	Offline

TABLE 4: Results of performance metrics for GA-Based Algorithms [12].

5. CONCLUSION AND FUTURE WORKS

In this paper a SVM based system is presented which can classify the IDS alerts with high accuracy and reduce number of false positive alerts considerably. Also the system is able to identify the attack types of the alerts more accurate in little time slice.

It seems to be useful using SVM to correlate alerts to discover attack sequences so this idea is a future work of this paper.

6. REFERENCES

[1] Debar, H., M. Dacier, and A. Wespi, *Towards a taxonomy of intrusion-detection systems.* Computer Networks, 1999. **31**(8): p. 805-822.

[2] Cortes, C. and V. Vapnik, Support-vector networks. Machine learning, 1995. 20(3): p. 273-297.

[3] Julisch, K., Clustering intrusion detection alarms to support root cause analysis. ACM Transactions on Information and System Security (TISSEC), 2003. 6(4): p. 443-471.

[4] Maheyzah, S.Z., Intelligent alert clustering model for network intrusion analysis. Journal in Advances Soft Computing and Its Applications (IJSCA), 2009. 1(1): p. 33-48.

[5] Wang, J., H. Wang, and G. Zhao. A GA-based Solution to an NP-hard Problem of Clustering Security Events. 2006. IEEE.

[6] DARPA 2000 Intrusion Detection Evaluation Datasets, M.L. Lab., Editor. 2000.

[7] Cuppens, F. Managing alerts in a multi-intrusion detection environment. 2001.

[8] MIRADOR, E. Mirador: a cooperative approach of IDS. in European Symposium on Research in Computer Security (ESORICS). 2000. Toulouse, France.

[9] Debar, H. and A. Wespi. Aggregation and Correlation of Intrusion-Detection Alerts. in Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection. 2001.

[10] Ahrabi, A.A.A., et al., A New System for Clustering and Classification of Intrusion Detection System Alerts Using Self-Organizing Maps. International Journal of Computer Science and Security (IJCSS), 2011. 4(6): p. 589.

[11] Kohonen, T., Self-Organized Maps. 1997, Science Berlin Heidelberg: Springer series in information.

[12] Bahrbegi, H., et al. A new system to evaluate GA-based clustering algorithms in Intrusion Detection alert management system. 2010. IEEE.

[13] Krovi, R. Genetic algorithms for clustering: a preliminary investigation. 1992. IEEE.

[14] Krishna, K. and M. Narasimha Murty, Genetic K-means algorithm. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 1999. 29(3): p. 433-439.

[15] Fuyan, L., C. Chouyong, and L. Shaoyi. An improved genetic approach. 2005. IEEE.

[16] Lu, Y., et al. FGKA: a fast genetic K-means clustering algorithm. 2004. ACM.

[17] Di Nuovo, A.G., V. Catania, and M. Palesi. The hybrid genetic fuzzy C-means: a reasoned implementation. in International Conference on Fuzzy Systems. 2006. World Scientific and Engineering Academy and Society (WSEAS).

[18] Ahrabi, A.A.A., et al., Using Learning Vector Quantization in IDS Alert Management System. International Journal of Computer Science and Security (IJCSS), 2012. 6(2): p. 1-7.

[19] Kohonen, T., Learning vector quantization, in M.A. Arbib (ed.), The Handbook of Brain Theory and Beural Networks. 1995: MIT Press.

[20] Njogu, H.W., et al., A comprehensive vulnerability based alert management approach for large networks. Future Generation Computer Systems, 2013. 29(1): p. 27-45.

[21] DARPA 1998 Intrusion Detection Evaluation Datasets, M.L. Lab., Editor. 2000.

[22] DARPA 1998 Intrusion Detection Evaluation Datasets, M.L. Lab., Editor. 1998.

[23] Brugger, S.T. and J. Chow, An Assessment of the DARPA IDS Evaluation Dataset Using Snort, D. UC Davis Technical Report CSE-2007-1, CA, Editor. 2007.

[24] Vapnik, V.N., The nature of statistical learning theory. 2000: Springer-Verlag New York Inc.

[25] Webb, A.R., Statistical pattern recognition. Second Edition ed. 2002, Malvern UK: Wiley.

[26] Matlab, www.mathworks.com/products/matlab/, Editor. 2009, Mathworks.

[27] Ma, J., Y. Zhao, and S. Ahalt, OSU SVM classifier matlab toolbox (ver 3.00). Pulsed Neural Networks, 2002.

A Review of Feature Model Position in the Software Product Line and Its Extraction Methods

Saba Pedram *pedram.saba@gmail.com*
Department of Computer
Islamic Azad University, Science and Research Branch
Tehran, Tehran, Iran

Mehran Mohsenzadeh *mohsenzadeh@srbiau.ac.ir*
Department of Computer
Islamic Azad University, Science and Research Branch
Tehran, Tehran, Iran

Amir Azimi Alasti Ahrabi *amir.azimi.alasti@gmail.com*
Department of Computer
Islamic Azad University, Shabestar Branch
Tabriz, East Azerbaijan, Iran

Abstract

The software has become a modern asset and competitive product. The product line that has long been used in manufacturing and construction industries nowadays has attracted a lot of attention in software industry. Most importance of product line engineering approach is in cost and time issues involved in marketing. Feature model is one of the most important methods of documenting variability in product line that shows product features and their dependencies. Because of the magnitude and complexity of the product line, build and maintain feature models are complex and time-consuming work. In this article feature model importance and position in product line is discussed and feature model extraction methods are reviewed and compared.

Keywords: Software Product Line, Feature Model, Extraction Method Review.

1. INTRODUCTION

The software has become a modern asset and competitive product [1]. The product line that has long been used in manufacturing and construction industries nowadays has attracted a lot of attention in software industry. Most importance of product line engineering approach is in cost and time issues involved in marketing; but product line engineering is also supports other business purposes [2].

According to the definition provided by Carnegie Mellon University, Software product line includes a family of system that have a series of technical assets in common between all of them (core assets) and variability parts that were considered in order to meet customers' specific requirements. Software product line engineering adds a lot of values for developer companies. Reusability, short time product presentation and quality are all of the aspects that make software product line development as a cost-effective approach [2].

Primarily, feature models were introduced in 1990 in feature-oriented domain analysis methods. FODA method support reusability in functional and architectural levels. Features are logical units of behavior that is described by a set of functionality and quality requirements and represent an important value for the users [3]. In features oriented design and implementation, feature models are standard visual presentation. Feature diagram describe features and integrity constraints between them as a tree. A feature generally indicates the abstract of a domain. But feature is more than a name in domain modeling; its other properties must be considered as well.

Some of the potential characteristics that domain experts can collect for the features as followed:

- Describes the features and their corresponding requirements
- Relationship with other features, especially the hierarchy, order and grouping
- The estimated or measurement cost of an achievement of a feature
- Configuration knowledge, such as “enabled by default”
- Constraints, such as “include feature X and exclude feature Y”
- Relationship between potential features

Features do not easily compose [4]. All features cannot be composed and some features may require the presence of other features. So feature model describes the relationship between the features and valid feature selections [5].

Implementation of the variability in the features level has an important effect on conceptual integrity of the system. Features used for domain modeling, variability management, guidance of feature planning and as the basis for communication between stakeholders in the system or used as general guidelines for the system design [6].

In order to emphasize the complexity of the design position in the product line, the following table compares product line approach and single software development briefly.

Criteria	Product line	Single system
Production area	Family of products	Single system
Development approach	Tow life cycle	Single life cycle
Reusability	Strategic and technical reusability	Technical reusability
Practical purpose	Meet the needs of multiple customers in certain segments of the market	Meet the needs of a customer
New product time to market	Short	Long
Development costs per system in long term	Decreasing (stable)	Increasing
Required architecture type	Reference architecture	Single system architecture
Variability	Numerous and mandatory	Limit
Requirement analysis complexity	High	Depend on product type high/low

TABLE 1: Compares product line approach and single software development.

The task of constructing feature models can be very arduous for requirements engineers, especially if they are presented with heterogeneous and unstructured requirements documents (such as interview transcripts, business models and technical specifications) from which to derive the feature model. The task of identifying core and variant features can be particularly difficult in the case of extractive or reactive SPLE [7].

In this article, feature model extraction methods are discussed. In section 2, importance and necessity of feature model in product line is reviewed from different viewpoint. Section 3 describes the feature model extraction methods. And in Section 4 feature model extraction methods have been compared.

2. FEATURE MODEL IMPORTANCE AND POSITION IN SOFTWARE PRODUCT LINE

Software product line engineering is an affordable approach for development of product family. Product line engineering approach focuses on the creation of reusable infrastructure. The key of its success is in having the correct view of the scope of software product line, complete identification of common variables, products and interconnections between the features. Software product line domain is analyzed by feature oriented domain analysis methods and modeled by feature model.

Features are the first product line engineering concerns. Describing feature is difficult since on the one hand is stakeholders and the other hand is design and implementation concepts [4]. At first glance, it seems that feature model is visual; but creation of good feature model that covers most of commonalities and variability and make reusable asset development easily, is difficult [8].

If the scope of the problem is determined to be incomplete, interrelated features may be not implemented, or implement features may not use. Such problems cause unnecessary complexity and increase development and maintaining costs [9]. To avoid these problems, software product line domain is usually modeled by feature model. Such as other model-based approaches, product line engineering faced with large-scale models and thousands uncommon features [10]. Construction and maintenance of such large models are very complex.

3. FEATURE MODEL EXTRACTION METHOD

The early roots of modeling features can be found in the architectural community. Architects use feature models for the production of simple and abstract overview of the architecture. The UML class diagram or package is an example of this application. Feature model describe commonalities and variables in the form of mandatory, alternative and optional form, also it highlights constraints. A basic feature model is a ‘and-or’ graph with some constraints for identifying valid product line configuration [11].

Feature model extract in different ways, in the following the methods are described briefly.

Mathieu Acher [12] used tabular format product description in their proposed feature model extraction method. This process is parameterized through a dedicated language and high-level directives (e.g., products/features scoping).in this method, several tabular data file that is collection of products description from different perspectives documents, is used for feature model extraction. They describe a specific merging algorithm that first computes the feature hierarchy and then synthesizes the variability information using propositional logic techniques. The main contributions of this paper are extraction process parameterized by a dedicated language and an automated procedure that synthesizes an FM based on product descriptions.

Darvil [1] proposed new approach for creating feature model from descriptions of products that are available on online repository of products and Web sites such as CNET softPedia. As each individual product descriptions show only a part of domain’s features, a large collection of descriptions can provide a fairly comprehensive coverage. In this method tow initial phase are introduced. One advantage of this method is availability of product description for each type of products to the public, it means that this method can also be used in an organization that has not developed the software for the target domain.

This [13] work presents an approach based on formal concept analysis that analyzes incidence matrices containing matching relations as input and creates feature models as output. New optimal approach presented here, do this conversion in a reasonable time even when the product is a high number. Incidence matrix is introduced as a new concept which describes the common and different artifacts of variants. Using this method leads to obtain feature model graph from conceptual graph directly.

Horatio [14] use mining techniques on public product description and use clustering algorithms for finding domain specific features and use probabilistic model. After they use association rules and k-means machine learning strategies for defining specific characteristics of a special product.

Mathieu [15] review the challenges of extracting a lot of feature models from the same input configuration that only a few of them are meaningful and can be maintained, so authors define a specific criteria for their separation. The challenge is to study various configuration models that are possible to obtain from one configuration, but only few of them are meaningful and sustainable. Various features model are identified primarily, and then key characteristics that distinguishing them is determined. Finally, with a public policy that is based on the understanding and knowledge of user, the feature model is constructed.

Krzysztof Czarnecki et al ‘s [16] approach introduce probability feature model and show how to obtain probability feature model from data mining techniques from feature configuration set. The author believes that the results are the basic foundation to build reverse engineering software product line tools.

Nathan Weston et al‘s method [7] introduce a tool suite which automatically processes natural-language requirements documents into a candidate feature model, which can be refined by the requirements engineer.

4. CONCLUSION

In order to compare reviewed methods, some criteria are proposed. These criteria are: customer requirement priority, product quality predictability, non-functional requirements selectivity, considering integrity constraint and Feature model extraction source.

Suggested criteria , customer requirement priority, considering integrity constraint are inspired from [17]. Product quality predictability and non-functional requirements selectivity are from [18] and 19]. Feature model extraction source is considered to compare method’s precondition and input.

- customer requirement priority [1], [12], [3] and [3] don’t have any priority. [20] do requirements priority in the form of user-defined functions that reflect the priorities of their target states.

- Product quality predictability: Since the product line include huge number of features and the number of products that can be produced with these features is growing exponentially, so the possibility of making all products and to measure their quality is far-fetched. The ability to predict the quality of individual products without making desirable products is desirable and effective manner in managing product line and marketing. Since and [15] and [1] extract feature model after product development, haven’t this ability. [3] have not expressed any predictive view for their quality products [20] provide a way to measure and predict the product's quality.

- Non-functional requirements selectivity: Non-functional requirements in the development of a single product collected and documented before development. During development, such as non-functional requirements frameworks tools, help the developers design decisions that affect the final product properties. But in product line approach, the product is designed and produced for a range of clients with different non-functional requirements. Sometimes different customers may need opposing non-functional requirements. So the ability of choosing non-functional features for each customer is a mean to enhancing product quality and meet customer demands. [3], [12], [7] don’t consider this capability. [20] method has this ability to predict product quality before product development by considering selected features.

- Considering integrity constraint: In addition to the parent-child relationship between the features, integrity constraints for domestic mutual dependencies between features are presented. The most important provisions of the integrity of the proposed are "include" and "exclude". [3], [1] and [12] don't use integrity constraint.in Nathan Weston method just use optional and alternative feature's constraints.

- Feature model extraction source: feature model extraction methods use different inputs for building feature model and cause differentiation between them. Source used in each way is presented in the table given below:

Feature model extraction source	**Method**
Public product description	Jean-Marc Davril,2013
Tabular data files	Mathieu Acher,2012
Customer requirements and common domain requirements	Conqueror,2011
Common domain requirements	FODA ,1990
Online public product description	Dumitru,2014
Possible configuration set	Acher, Baudry, 2013
Data mining techniques from feature configurations	Czarnecki, She,2008
Using (RDF) Request Defining Language and requirement definition	Nathan Weston, 2009

TABLE 2: Comparison extraction's model feature source.

Although a large variety of different methods have been proposed to extract feature model in product line, the development team must use appropriate method based on their emerging needs and their available inputs.

5. FUTURE WORK

A large variety of different feature model extraction methods have been proposed in software product line. There is the possibility of converting feature model to product line architecture. We will do future research on how to convert feature model to software product line architecture.

6. REFERENCES

[1] Davril, J.-M., et al. *Feature model extraction from large collections of informal product descriptions*. in *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*. 2013. ACM.

[2] Van der Linden, F.J., K. Schmid, and E. Rommes, *Software product lines in action: the best industrial practice in product line engineering*. 2007: Springer Science & Business Media.

[3] Kang, K.C., et al., *Feature-oriented domain analysis (FODA) feasibility study*. 1990, DTIC Document.

[4] Apel, S., et al., *Feature-Oriented Software Product Lines*. 2013: Springer.

[5] Matinlassi, M. *Comparison of software product line architecture design methods: COPA, FAST, FORM, KobrA and QADA*. in *Proceedings of the 26th International Conference on Software Engineering*. 2004. IEEE Computer Society.

[6] Sochos, P., M. Riebisch, and I. Philippow. *The feature-architecture mapping (farm) method for feature-oriented development of software product lines*. in *Engineering of Computer*

Based Systems, 2006. ECBS 2006. 13th Annual IEEE International Symposium and Workshop on. 2006. IEEE.

[7] Weston, N., R. Chitchyan, and A. Rashid. *A framework for constructing semantically composable feature models from natural language requirements*. in *Proceedings of the 13th International Software Product Line Conference*. 2009. Carnegie Mellon University.

[8] Pleuss, A., et al., *Model-driven support for product line evolution on feature level.* Journal of Systems and Software, 2012. **85**(10): p. 2261-2274.

[9] Heradio-Gil, R., et al., *Supporting commonality-based analysis of software product lines.* IET software, 2011. **5**(6): p. 496-509.

[10] Acher, M., et al., *Reverse engineering architectural feature models*, in *Software Architecture*. 2011, Springer. p. 220-235.

[11] Acher, M., et al., *Familiar: A domain-specific language for large scale management of feature models.* Science of Computer Programming, 2013. **78**(6): p. 657-681.

[12] Acher, M., et al. *On extracting feature models from product descriptions*. in *Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems*. 2012. ACM.

[13] Ryssel, U., J. Ploennigs, and K. Kabitzsch. *Extraction of feature models from formal contexts*. in *Proceedings of the 15th International Software Product Line Conference, Volume 2*. 2011. ACM.

[14] Dumitru, H., et al. *On-demand feature recommendations derived from mining public product descriptions*. in *Software Engineering (ICSE), 2011 33rd International Conference on*. 2011. IEEE.

[15] Acher, M., et al. *Support for reverse engineering and maintaining feature models*. in *Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems*. 2013. ACM.

[16] Czarnecki, K., S. She, and A. Wasowski. *Sample spaces and feature models: There and back again*. in *Software Product Line Conference, 2008. SPLC'08. 12th International*. 2008. IEEE.

[17] Asadi, M., et al., *The effects of visualization and interaction techniques on feature model configuration.* Empirical Software Engineering, 2014: p. 1-38.

[18] Sincero, J., W. Schroder-Preikschat, and O. Spinczyk. *Approaching non-functional properties of software product lines: Learning from products*. in *Software Engineering Conference (APSEC), 2010 17th Asia Pacific*. 2010. IEEE.

[19] Siegmund, N., et al. *Scalable prediction of non-functional properties in software product lines*. in *Software Product Line Conference (SPLC), 2011 15th International*. 2011. IEEE.

[20] Siegmund, N., et al., *SPL Conqueror: Toward optimization of non-functional properties in software product lines.* Software Quality Journal, 2012. **20**(3-4): p. 487-517.

INSTRUCTIONS TO CONTRIBUTORS

The *International Journal of Computer Science and Security (IJCSS)* is a refereed online journal which is a forum for publication of current research in computer science and computer security technologies. It considers any material dealing primarily with the technological aspects of computer science and computer security. The journal is targeted to be read by academics, scholars, advanced students, practitioners, and those seeking an update on current experience and future prospects in relation to all aspects computer science in general but specific to computer security themes. Subjects covered include: access control, computer security, cryptography, communications and data security, databases, electronic commerce, multimedia, bioinformatics, signal processing and image processing etc.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCSS.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Started with Volume 9, 2015, IJCSS is appearing with more focused issues. Besides normal publications, IJCSS intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

We are open to contributions, proposals for any topic as well as for editors and reviewers. We understand that it is through the effort of volunteers that CSC Journals continues to grow and flourish.

IJCSS LIST OF TOPICS
The realm of International Journal of Computer Science and Security (IJCSS) extends, but not limited, to the following:

- Authentication and authorization models
- Computer Engineering
- Computer Networks
- Cryptography
- Databases
- Image processing
- Operating systems
- Programming languages
- Signal processing
- Theory
- Communications and data security
- Bioinformatics
- Computer graphics
- Computer security
- Data mining
- Electronic commerce
- Object Orientation
- Parallel and distributed processing
- Robotics
- Software engineering

CALL FOR PAPERS

Volume: 9 - **Issue: 6**

i. Submission Deadline : October 31, 2015 **ii. Author Notification:** November 30, 2015

iii. Issue Publication: December 2015

CONTACT INFORMATION

Computer Science Journals Sdn BhD

B-5-8 Plaza Mont Kiara, Mont Kiara

50480, Kuala Lumpur, MALAYSIA

Phone: 006 03 6204 5627
Fax: 006 03 6204 5628
Email: cscpress@cscjournals.org

CSC PUBLISHERS © 2015
COMPUTER SCIENCE JOURNALS SDN BHD
B-5-8 PLAZA MONT KIARA
MONT KIARA
50480, KUALA LUMPUR
MALAYSIA

PHONE: 006 03 6204 5627
FAX: 006 03 6204 5628
EMAIL: cscpress@cscjournals.org