CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Chapter two discusses the key definition and concept employed throughout this research. Accordingly, the focus of this chapter is on the identification and evaluation of the existing works related with cloud worm attacks. The main content in this chapter acts as background information as well as a reference for the discussion of the findings of this research. This chapter explores cloud computing definition, implication of worm attacks in cloud environment, and the detection and response methods of these attacks. This is because cloud infrastructure is connected to the network, in which, cloud worm is able to spread and propagate within the network, and therefore, harming the whole cloud infrastructure (Waston, 2012; Zhang *et al.*, 2010).

2.2 Cloud Computing

Cloud computing has been explained as a form of service which delivers information with the use of technology (Carlin & Curran, 2011). Cloud can also be explained as a form of technology that comprises a combination of applications, services as well as an enormous database that is able to handle large number of computers over a shared network (McFedries, 2008). Parekh and Sridaran (2013) on the other hand, defined cloud as a form of support services to store and access large forms of data and resources. Cloud application employs the use of the internet in order to provide accessibility between service providers and clients in order that both could communicate with ease as well as having access to various services (Carlin & Curran, 2011).

As for this research, cloud computing is perceived as a technology that is able to provide Internet services with applications and resources which allow users to be able

to access and store a large amount of data by using web services anytime and anywhere.

2.2.1 Cloud computing characteristics

A total of five different characteristics are defined as cloud services according to Carlin and Curran (2011). The first characteristic is that; cloud service uses shared resources among different users. This allows different users to share the same application from the web at different times. The second characteristic of cloud is that; it has a huge scale of accessibility. Cloud service offers the ability to use thousands of different systems and applications, with huge amounts of storage space as well as bandwidth to its users. Moreover, cloud also provides flexibility to its users, enabling them to increase or decrease the amount of resources required, making it more organised and easily accessible for the users. Cloud users have the freedom to pay for resources that they are using or have used, which saves money as they do not have to pay for the resources that are not used. Furthermore, cloud users also have the opportunity of self-provisioning of resources that are required, such as additional network resources or physical resources, including storage, when required (Mather *et al.*, 2009).

2.2.2 Cloud service models

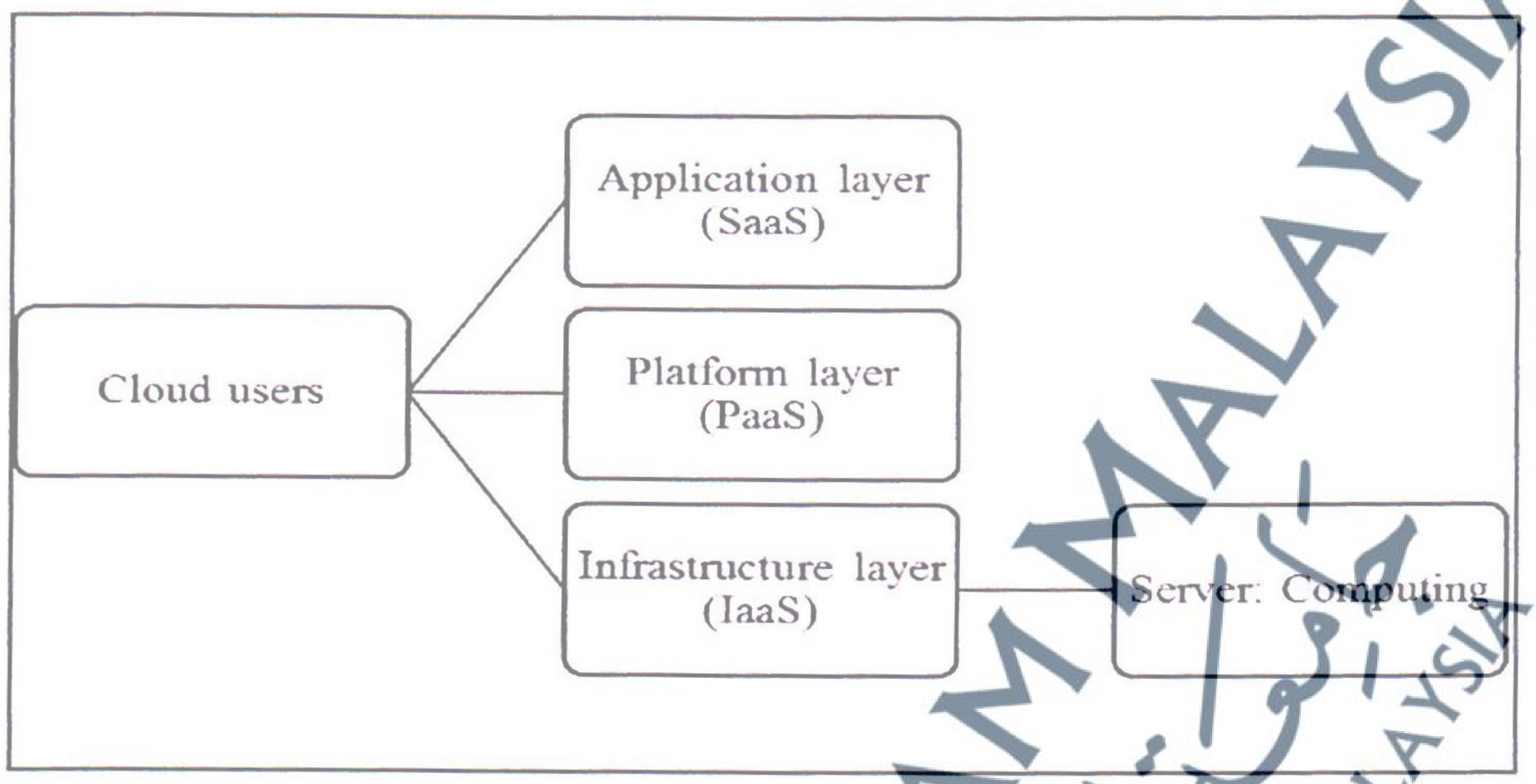
There are three types of services that are offered by cloud computing (Sowmya *et al.*, 2014; Freet *et al.*, 2015; Gupta, 2015; Hwang *et al.*, 2016; Noor *et al.*, 2016) as shown in Figure 2.1, which are:

I. Software as a service (SaaS): Cloud service provider runs the services and delivers accessibility to its users through the internet. These services are usually used by organisations and individuals.

II. Platform as a Service (PaaS): PaaS provides a centralised operating system, whereby it provides access to process, execution, and running of software and applications.

III. Infrastructure as a Service (IaaS): This form of service is handled and managed by cloud providers that are in charge of different operations including

networking and hardware management. Physical machines or servers can be rented through this service.



Source: (Sowmya *et al.*, 2014)

Figure 2.1: Cloud service models

2.2.3 Cloud deployment model

According to several researchers (e.g., Mell & Grance, 2011; Parekh & Sridaran, 2013; Carlin & Curran, 2011), cloud deployment models are focused on four different areas which are: public, private, hybrid and community clouds. These are explained in detail as the following:

(i) *Public clouds*: These are among the most used clouds, in which, multiple users are able to have access to web and service applications over the internet. Individual users are going to have access to their individual resources which are often hosted by separate cloud service providers with many data centres.

(ii) *Private clouds*: Users of private cloud will be on a private network whereby they will have access to data and services in a controlled network. This form of cloud environment provides users with the supervised process of data management as well as tighter security measures.

(iii) *Hybrid clouds*: Hybrid cloud system incorporates the public and private cloud features within the same network. This provides advantages for organisations; they could benefit from both sets of cloud features. For instance, an organisation can have control over confidential and private information on the private cloud system, while the general public are still able to use public cloud to have access to large traffic.

(iv) *Community cloud*: Cloud computing also has another model, which is the community model. This model can be shared by a number of different organisations to facilitate a particular community that shares common interests or concerns. This type of cloud is often handled by a service provider and is often available whether the user is on the premises or off the premises.

As for this research, the main focus is to develop a cloud worm detection and response model that can adapt to all types of cloud computing environment. Private cloud is considered as more secured but it could be affected by internal or authorised users (Modi *et al.*, 2013). More thread could be affected by public cloud because it is open for all users through internet. Even hybrid cloud is exposed to worm attacks because it is the combination of public and private cloud. Therefore, this research helps to improve security by developing a new cloud worm detection and response technique.

2.3 Worms attack in cloud computing

Worm is explained as a piece of malicious code that replicates itself and spreads throughout computer networks (Agarwal *et al.*, 2014). Also, worm is explained as a malicious code that duplicates itself and can automatically transfer from one infected machine to the other through a network without any action by the user (Saudi, 2011). Additionally, in cloud computing, worm attacks can be serious due to the large scale of data and users involved. Cloud worm is known as a malicious code that has the ability to propagate and replicate itself which causes damage, interrupts or stops cloud resource or services (Biedermann & Katzenbeisser, 2012).

According to Zunnurhain and Vrbsky (2010), the worm injection attacks launched in a cloud environment makes the worms form a real service operation in cloud system.

The worm is often injected by a malicious user or through a malicious service, into cloud system. As a result of this, cloud's data and information becomes corrupted, making it difficult to use and access. In some cases of worm attacks, users are forced to wait until the attack is over, since the worms modifies the data in cloud, which causes deadlocks. This does not allow for any user to use the system.

Qaisar and Khawaja (2012) also mentioned that worm attacks occur when worms are injected into a computer application or system. Here, the attacker's main motive is to create a personal spiteful attack. Hence, cloud environment would be damaged in terms of service, application or through virtual machine. If the worm attack is successful, users will request for the use of the spiteful service as they would not know that it is a malicious service. This makes it easier for the attacker to insert the spiteful code into the system. Following this, the attacker may then pursue to upload different viruses in the form of programs, whereby they are spread across cloud applications and structures. These instances show a need for organisations to be able to detect worm attacks and handle these attacks before it destroys information in the system. Therefore, it is necessary to look at different worm detection models before developing a new and more efficient worm model.

Another study by Mishra (2014) described that cloud worm attack is spreading rapidly these days through the network, and many cloud services have been affected. The method of this attack is initiated by compromising FTP to sniff password from cloud system and send it back to the hacker. Then, the hacker applies the collected FTP password to get access to cloud services and virtual machine. In this case, the hacker can exploit its privileged access capabilities to the service instances in order to attack the service instance's security domains. This way, hacker can inject his malicious code and infect other cloud users who work with the same virtual machine or services. This causes blocking of the services, and users are not able to get into cloud services. Also, the hacker has the ability to gain control over the user's data in cloud system.

Meanwhile, hypervisor is a type of software which grants multiple guest VMs to run simultaneously at the distinct server. It is liable to implement isolation between VMs and hardware resource management (Arya *et al.*, 2013). Additionally, it can control

the access to the VMs and physical resources. Once hypervisor is attacked by the worm, it affects all the VMs because all VM operations become traced unencrypted (Bouayad *et al.*, 2012). As for worm attack on VM in cloud, the worm might benefit from the vulnerability in the virtual machines to spread its malicious code and damage as much as possible within the virtual machines in cloud (Bharadwaja *et al.*, 2011). Hence, hypervisor needs to be carefully monitored to prevent worm attacks (Arya *et al.*, 2013). Worm injection is an attack method where worm attaches malicious code into cloud applications to gain access to user's information, cloud resources, or databases. The worm then performs tasks like stealing personal information, changing data, or further spreading the fake code (Duncan *et al.*, 2015; Sgandurra & Lupu, 2016).

According to Stephanakis *et al.*, (2015) worm could initiate web service attack in cloud where it scans all the ports of virtual machines and host machines in order to find open port. Then, the worm can take advantage of those open service ports. Therefore, hypervisor should not grant vulnerable or malicious machines in its environment.

2.3.1 Issues related with cloud worm attacks

One of the vulnerabilities caused by cloud worm attacks causes the nefarious and abuse of cloud computing, due to relative anonymity of cloud subscription. Worm attacks caused by malicious groups may use cloud space as platform for launching their attacks. Also, another vulnerability caused by cloud worm attacks is through virtual networks which are connected with virtual circuits in cloud. Through unregulated data backup, cloud worm attacks could lead to unscheduled system rollback which may result in the resetting of known threats and also lack of policy regulation during server migration. Cloud worm attacks are also able to infect hypervisor programs, leading to new vulnerability within cloud. Additionally, cloud worm attack infects the virtualised environment which also leads to data loss and leakages (Hashizume *et al.*, 2013; Watson, 2012; Archer *et al.*, 2010; Wu *et al.*, 2010). In this case, once cloud worm affects the virtual machine and hypervisor, it incurs more damages and can control the whole cloud at the same time (Nalinipriya *et al.*, 2016; Sgandurra & Lupu, 2016).

Another issue caused by cloud worm attacks is the challenges they cause for cloud computing models. Cloud worms affects cloud service models such as SaaS, PaaS and IaaS which may lead to loss of personal and confidential information, and data breach (Parekh & Sridaran, 2013).

Cloud worm attacks are also able to easily find and access the IP addresses stored in cloud servers. Thus, it will be easier for malicious users to plan an attack using the location of the user leading to data copying and saving. As a result, malicious user could access the data even after cloud service provider changes or deletes its data due to the attack. This causes long term impacts on cloud users as their data will still be accessible and they may still be at risks. In addition, cloud-based applications provide hackers with the opportunity to host a worm application owing to the fact that cloud service providers offer their services to any legal user. This allows hacker to easily cause worm DDoS attack against cloud provider or through the hacking of another cloud user's account (Sabahi, 2011).

Cloud worm has also caused numerous security threats in different ways. One of these security threats by cloud worm is that it attacks the initialisation process during VM creation. It does this by injecting VM image as a worm and stores itself within the repository, causing threats to the VM structure. Another threat caused by cloud worm is the denial of service attack due to overloading of the system resources with junk processes which initiates on the physical system, leading to instability in cloud system (Hashizume *et al.*, 2013).

Another threat caused by cloud worm is the snoop attack which takes place during VM migration process which puts the network at risk. This is because data can be accessed secretly and stored into an unknown host during the migration (Sun *et al.*, 2016). Cloud worm attacks can also lead to user account hijacking through weak password choices including social engineering. Such hijacking leads to the stealing of sensitive and financial information of user (Kazim *et al.*, 2015).

Worm is a big threat for cloud computing and it is handled by malicious codes. These worms follow various strategies to attack cloud and challenge the security of cloud. By classifying cloud worm attacks and behaviours, threats could be minimised to make cloud more secure.

2.4 Comparison of worm attacks with other malicious attacks in cloud

There are different types of malicious attacks that are facing cloud computing in recent times. With the coming of these threats, cloud computing networks has been witnessing a lot of security issues that have been causing many types of distortions which relate to either stealing online files, slowing down of networks and many more problems. Such types of these significant malicious attacks are explained as follows:

i. Side Channel Attacks: - Side channel attacks employ two steps known as VM CO-Residence and Placement. The attacker regularly places his instance within the physical machine or device as a VM Extraction. This makes it possible for the malicious instance to use side channels to obtain and learn information from a device. The attacker tries to take over cloud system through inserting a malicious virtual machine close to a targeted cloud system and commence the side channel attack (Chouhan & Singh, 2016).

ii. Denial-of-Service attack (DoS): - Cloud is known to be weaker than DoS attacks because of many users that are using cloud service and resources, which causes the Dos attack to be damaging. Cloud computing operating system begins to operate with more computational power. In this process, the server hardware boundaries begin to restrict. In this situation, cloud system protects itself by increasing its computational power thereby, helping to do most damage to the service availability through single attack entry point (Chou, 2013; Singh *et al.*, 2014).

iii. Authentication Attack: - Authentication is known to be a weak point in cloud computing services which is repeatedly targeted by an attacker. In recent times, most of the services still use simple username and password for authentication. These attacks consist of Brute Force Attacks which are used by

attackers to break passwords. The success of this attack usually depends on how powerful the computing capability is. This is because thousands of possible passwords are needed to be sent to a target user's account in cloud until it finds the correct one to access. Another attack related to authentication attack is the replay attacks. This attack which is also known as playback attack is a form of network attack in cloud in which a valid data transmission is maliciously repeated or delayed. The next attack related to authentication attack is Shoulder Surfing. This is a substitute name of "spying" in which the attacker spies on the user's movements to get his password. In this type of attack, the attacker observes the user; how he enters the password for example, what kind of keyboard the user has pressed (Chouhan & Singh, 2016).

iv. Theft of Service Attacks:- The Theft of service attack makes use of vulnerabilities in the scheduler of hypervisors. This attack is carried out once the hypervisor uses a scheduling mechanism; in this case, it fails to detect the account of Central Processing Unit (CPU) usage caused by poorly behaved virtual machines. This failure can allow malicious users to have access to cloud services. This attack is more related to the public clouds where users are charged by the amount of time their VM is running instead of by the amount of CPU time used (Zhou *et al.*, 2013).

v. Phishing Attacks: - It is known to be an offensive behaviour in which an attacker hosts a phishing attack site on cloud by using one of cloud services and hijack accounts and services in cloud through social engineering techniques. The purpose of Phishing is to try to have access to personal information from unsuspecting victim by social engineering techniques. It is usually achieved by sending webpages links through emails. These links look like an authentic site, that is, a legitimate site such as credit card information or bank account login, but in this case, user is taken to fake locations. Through this trick, the attacker can get sensitive information such as passwords and credit card information from the target user (Khalil *et al.*, 2014).

vi. Cloud Worm Attacks: - Worm attacks in cloud are located in the networks where the worm intrudes. Here, the worm looks like an authentic service operation thereby making cloud services unethical. Cloud worm attacks are complex and can change data or results significantly in a gridlock or congestion which forces cloud user to wait until the attacks are concluded. The worm operates in cloud computing operations and obtains rights and concessions to control all cloud surroundings (Marhusin *et al.*, 2015).

vii. Worms can be a serious threat to cloud computing and network due to some characteristics it may possess. According to Shahine (2014) and Biedermann (2012), some worms use some stealth techniques to hide themselves on the cloud, making it difficult to be detected; it can also hide the process running within the cloud. Worms can also hide cloud user files and it can also delete the logs. Other worms that can infect cloud include Polymorphic worms (Shahine, 2014; Biedermann, 2012). This type of worm has the ability to change itself during its propagation in cloud. It makes signature based detection difficult to trace. As for the file worm, it is known as a modified version of virus but it does not connect its presence with any executable files within the cloud. It simply copies its code to a directory within the cloud expecting its new copies to be someday executed by the victim or cloud user. Lastly, for the multi-vector worm, it uses different type of propagation method within the cloud in order to make or create more duplicates into the cloud that are vulnerable for attack and effectively propagate behind the firewalls.

The comparison between worm attacks and other malicious attacks in cloud is summarised in the table below:

Table 2.1: Comparison of worm attacks with other malicious attacks in cloud

Various Malware Attacks	**Impact**	**Affect Confidentiality, Integrity and Availability (CIA)**
Denial of Service Attacks	• Flooding of cloud network • Loss of availability on cloud services.	• Availability

Side Channel Attacks	• Insert malicious VM • Facilitate the attainment of secret information from a device	• Confidentiality
Authentication Attacks	• Steals username and passwords • Data transmission is maliciously repeated	• Confidentiality
Theft of Service Attacks	• Allow malicious users to have access to cloud services through hypervisor failure	• Confidentiality
Phishing Attacks	• Gain access to personal information within cloud • Hijack accounts and services in cloud	• Confidentiality
Cloud Worm Attacks	• Used to steal sensitive data in cloud • Hide user files and delete cloud logs. • Change data significantly or results in a congestion which forces user to wait until the attacks are concluded • Alter or destroy files, transmit passwords, or leave copies of itself (Through payload)	• Confidentiality, Integrity and Availability

This research concentrates on choosing worm malicious attack due to the fact that cloud worms are serious vindictive codes that can seriously destroy an application and system in cloud environment (Chou, 2013). Worms attacks against cloud computing software and platform services have also become increasingly alarming in recent times. It is a malicious code that can distort the functionality of cloud and also can minimise cloud's system activity by attacking the virtual and hypervisor machine. These views are also shared by different researchers in the same field; these researchers have being trying to come up with other ways to tackle such problems. These views are shared in write ups by researchers such as Singh *et al.*, in 2014, Watson et al. in 2015 and Qaisar and Khawaja in 2012. For more explanation on CIA, refer to page 53, section 2.13.

2.5 Cloud Worm Classification

Classification is a systematic arrangement in groups or categories according to established criteria. In the aspect of worm classification, it shows that classification is the systematic arrangement of worm characteristics and sub-categories into groups in order to help measure the efficiency of detection (Pratama & Rafrastara, 2012). Worm classification can also be based on its structure. These structures include infection propagation. One of the features discovered in worm's structure shows its capability to take control of a remote system which it does by transferring itself to a new file path within the system. It shows that the creator of the worm may use script language or any document to damage a computer system which includes Remote Control and update Interface. This is known to be another feature process utilised by the worm. Worm is also known to use the communication module which is the vital part of remote control in order for the author to control the worm through sending of control messages to the worm copies. Secondly, life-cycle manager is a feature where the worm is run by its author for a pre-set period of time. It has been discovered that the worm contains bugs and continues to run repeatedly. Another one is payload. As for the payload feature, the activity in the way it propagates is usually different from the code itself. This feature totally depends on the attacker's reasons and imagination, which shows that different payloads may have different ways of reaching ends directly. Lastly, self-tracking is a feature known to be usually sent in order to track any system path. The worm sends information by e-mail regarding an infected computer to monitor its spread. This write-up have aided in gaining better understanding on worm classification together with the worm itself, by showing how they behave and how to counter the worms' behaviour. However, the experiment carried out by these authors was limited to the PC environment, which focuses on the computer itself.

Rajesh *et al.* (2015) also explained different types of worm classification with their unique features. In their own research, they classified the computer worms in two categories. The first is classification based on behaviour and the second is classification based on scanning. For the worm classification that is based on behaviour, it is listed in five parts which are stealth worms, Polymorphic worms, File

worms, Multi-vector worms and Email worms. As for the worm classification that is based on scanning, the features are listed as Random scanning, Localised scanning, Sequential scanning, Topological scanning and Hit list scanning. Somehow, similar to other research, this study is also limited to PC and the computer networks. In this article, the writers were able to do a study on how worms are evolving including the damages they are able to cause to networks together with their classification features. Their study has proven how dangerous these worms are to the computer world.

Also, according to a study by Saudi (2011), worm classification can be used as the basis for a worm detection and response technique. Her study was able to do a STAKCERT worm classification based on the testing and comparison associated with the research by Dabirsiaghi (2008), Nazario *et al.*, (2001), Helenius (2002), Skoudis and Zelster (2004) and Saudi *et al.*, (2008a). In addition, an article written by Saudi *et al.*, (2008) involves an analysis which is based on type of worm classification feature known as EDOWA. They elaborated on the types of worm classification in different features. First of all, they explained that Infection is a worm classification feature that shows how a computer gets infected. The worm gets initial control of the computer system. The authors stated that a worm infects a host primarily in two ways: either by an error within running software or by action that is carried out by the user. All these could be in the form of either a host or a network. The next feature is Activation. As for the Activation classification feature, it was described as a trigger mechanism from the worm itself. Other researchers stated that the worm usually launches an attack on an identified targeted system which could be in the form of either "No Activation Phase," "Human Trigger," "Schedule Process," "Self-Activation Phase" or "Hybrid Lunch." Another feature is Payload. As for this part of the worm classification feature, payload is defined as a code with destructive capability. The feature found in this worm is not just limited to openly spreading mechanism. Rather, it usually depends on the kind of payload worm used which is stated as No Payload, Installing backdoor, Denial of Service, Destructive, and Phishing. As for Operation Algorithms, this article explained that this classification feature is used in avoiding detection. Accordingly, research that used operating algorithms as its classification was carried out by Albanese *et al.*, (2004), in which they classified worm as a survival. As explained by the authors, this classification feature attempts to avoid detection categorised as

Polymorphic and Stealth. For the research stated in this article which worked on producing the EDOWA system, they also made use of these five classification features. The paper also stated that these classification features can be used as a basis for worm classification or other upcoming research. However, this research is also limited to the execution of experiment within computers and networks.

Suleiman and Husain (2015) provided a general architecture of computer malware, and in this study, how computer malware affects computing devices is broken down into four different categories. These malware categories include the infection mechanism used. Regarding this feature, they explained that the computer malware tries to locate new host devices to infect through different procedures. It is carried out in ways such as random scanning, permutation scanning, localised scanning, hit list scanning, topological scanning, metaserver scanning and passive scanning. Another one is called propagation mechanism employed. For this malware classification feature, most of its infection process is to keep spreading and transmitting itself within a system in numerous ways which could be through mechanisms such as Self-carried propagation, embedded propagation and Secondary channel propagation. Activation mechanism used was another feature in their work. For this malware category, it involves an activation or trigger activity in order to show its behaviour within an infected computer host. The list of these activities includes Human activation, Activated by schedule process, and Self Activation. Regarding the nature of attack, the feature of this category shows different behaviours that a system might exhibit after being attacked. Some malwares show an immediate effect after infecting a host while others remain dormant or hidden until the damages are being done to the system. The nature of attacks shown by such malwares includes encryption of files, data theft, modifying and controlling. Accordingly, Suleiman and Husain's (2015) article was able to explain the numerous phases, showing the complete architecture of computer malware. The article was also able to describe the nature of the attacks by the computer malware by way of either transmission media, nature of damage or intelligence. However, the research related to this article is only focused or limited to the PC networks and memory.

Weaver *et al.*, (2003) explained that worms have five basic features known as infection, activation, payload, and propagation and operating algorithm. In addition, targeted discovery was explained. This feature shows that before the worm infects a computer device, it has to locate or discover an existing machine. Further, the worms are able to locate new machines through different processes which are scanning, external target list, pre-generated target list, internal target list, and passive monitoring. Another explanation is propagation carriers and distribution mechanisms. Under this feature, it shows how the level of propagation that takes place can affect the level of stealth and speed of a worm. It shows that the worm is either able to spread or propagate by itself or it can be transferred through the communication path. These processes are known as self-carried, second channel and embedded process. The next feature explained is activation. This classification feature shows that some worms can be activated immediately. On the other hand, other worms might take longer periods which could be in days or weeks before being activated. The type of activation is known as Human Activation, Human Activity-Based activation, Schedule process activation and Self activation. The next feature is Payload. As for this type of worm feature, it depends on the imagination and the intention of the attacker. Different attackers will come up with different design of their payload in order to fulfil their different needs or ends. Some of the types of payloads that they might use include non-functional, Internet remote control, Spam-Relays and Internet DOS. The last feature is Motivation and Attackers. For this aspect of worm attacks, it is very vital to have the idea of what motivates such attacks and find out those responsible for such attacks. Such reasons for these motivations may include Experimental curiosity, Pride and power, Commercial advantage, Extortion and criminal gain, and Random protest. Regarding this research, the authors were able to come up with taxonomy of worms based on the points listed above. This article could assist those who want to come up with a better defence to counter how the attackers design such malicious wares. As for this article's scope of this study, it also does not go beyond the computer network and their memory.

Although worms have those basic characteristics, each of these characteristics can be varied based on the network environment and domain. Based on these basic features of worms, and using dynamic analysis tools and experimentation, a new cloud worm

classification is to be introduced in this research. Cloud worm classification can be utilised as a basis for worm detection and response method which is to help increase the accuracy of detection rate. More detailed classification can be referred in chapter 4, section 4.3.1. The summarisation of worm's features can be seen in the table below.

Table 2.2: Summary of Worm Classification

Paper Titles	Worm Features	Strength	Weakness
Computer Worm Classification (Pratama and Rafrastsra , 2012)	• Infection Propagation • Remote Control and update Interface • Life-Cycle Manager • Payload • Self-Tracking	• Fosters better understanding regarding worm classification • Shows how worms behave • Counters the worms' behaviour	Limited to only PC memory
A Taxonomy of Computer Worms(Weaver *et al.*,2003)	• Motivation and Attackers • Payloads: • Activation • Propagation carriers and distribution mechanisms • Targeted Discovery	• Assists those who want to come up with a better defence	Limited to only PC memory
A Survey Paper on Malicious Computer Worms (Rajesh *et al.*, 2015)	• Classification based on scanning • Classification based on behaviour	• Describes how worms evolve including the damages they inflict to networks • Proves how worms are causing damages to the computer world	Limited to only PC memory
EDOWA Worm Classification (Saudi *et al.*, 2008)	• Infection • Activation • Payload • Operation Algorithms	• Presents classification features that can be used as a basis for worm classification in other upcoming research	Limited to only PC memory
Study of Computer Malware and Its Taxonomy (Suleiman and Husain, 2015)	• Infection Mechanism used • Propagation/Spreading Mechanism employed • Activation mechanism used • Nature of Attack	• Shows the complete architecture on computer malware • described the nature of the attacks by the computer malware	Limited to only PC memory

For the existing works that were mentioned above, improvement is to be made which is meant to improve the protection for cloud system. By introducing of the new EGA system in this research, a new cloud worm classification is proposed and it can be utilised as a basis for worm detection and response method. Further, this study also attempts to improve accuracy in cloud worm detection including the infection part which is one of the distinct features of this study, which will contribute to the mechanism of new cloud worm classification. Additionally, most of the past works were mostly focusing on the PC environment and their related network. Hence, this research is focusing on cloud computing.

2.6 Cloud Worm Analysis Techniques

Most of the techniques used for worm detection can be categorised into static and dynamic analysis. The worm analysis methods aid the analyst in understanding the risks related to a malicious code sample. The knowledge to be obtained can be used to respond to new trends in worm development or in taking the safety precaution to cope with the incoming future threats. Features derived from analysis of worm can be used to put together unknown worm and classify them into their existing families (Gandotra *et al.*, 2014).

2.6.1 Cloud Worm Dynamic Analysis

Analysing the behaviour of a cloud worm that deals with interaction of the system, while it is being executed in a controlled environment, which could be a virtual machine, simulator and emulator, is called dynamic analysis. Before executing cloud worm, the proper monitoring tools like Process Monitor used for file system and registry monitoring, and Process Explorer (for process monitoring), Wireshark (for network monitoring), are installed and activated. Dynamic analysis is more effective as compared to static analysis, and it does not require the executable to be disassembled. Dynamic analysis discloses the malwares' natural behaviour which is more resilient to static analysis (Gandotra *et al.*, 2014).

Dynamic analysis is also known to be a process of executing the sample, often in a virtual environment. It records the changes made during the execution time and

monitors the analysis environment. Dynamic features are those parts that are extracted from the recorded changes of dynamic analysis. Even though dynamic analysis has long analysis method, it handles many of the limits found in static analysis which enables it to extract features from packed and obfuscated worms which are not able to be handled by static analysis (Kumar *et al.*, 2016). Examples of information that can be obtained by dynamic analysis include API calls, system calls, instruction traces, registry changes, memory writes, and so on.

Numerous online automated tools exist for dynamic analysis of worms such as Virus total (Zolkipli *et al.*, 2011; Rieck *et al.*, 2011; Balazs, 2016; Akour *et al.*, 2016). This tool also can automatically analyse and also generate the behaviour reports of the worm (Firdausi *et al.*, 2010). Also, there are many researchers who used dynamic analysis in their works to conduct their experiment (e.g., Zolkipli *et al.*, 2011; Park *et al.*, 2013; Qiao *et al.*, 2013; Nari *et al.*, 2013).

As for static analysis, it analyses worm without executing. For this reason, it is not to be used in this research due to the fact that static analysis is only able to detect the known worm signature accurately which may sometimes fail in analysing unknown worm signature in its database because it is normally signature based which needs to be updated regularly, while also needing human experts to create the new signature (Kuber, 2014; Bazrafshan, 2013; Elhadi *et al.*, 2012).

Dynamic analysis is used for this research because of its capability and reliability based on previous studies by other researchers. Also, it is evident that dynamic analysis can be viewed as a good source of information on worm behaviour. An accurate model can be obtained from dynamic analysis (Damodaran, 2015). More detailed explanation can be referred in chapter 3, section 3.3.6.3.

2.7 Machine Learning

In recent years, the interest to use machine learning tools for malware analysis has been on the rise. These tools are considered important and powerful when dealing with huge datasets. Machine learning consists of algorithms that help different systems in learning and replicating any natural system. In this procedure, data is used

to train the system and help it behave in the same way with the environment of the natural system that it is going to copy (Armstrong, 2015). Machine learning is divided into supervised and unsupervised learning. Both of these techniques are to be applied in this research. These are explained as follows.

2.7.1 Supervised Learning

In supervised learning, each observed object is assigned to a label according to its characteristics by an algorithm. A class of labels is provided by the supervising variable for each classification problem. Hence, the feature extractor stores the feature vectors in order to assign objects to their suitable class. This process is governed by training the classifier to use the feature vector. The main challenge arises when the classifier assigns the object incorrectly, as the classifier compares the object according to the defined class and assigns it to the closest class. Hence, a similarity measure, to a reference object for that class will be done in order to validate this class (Berg, 2011).

In this research, supervised learning is also to be implemented in order to classify the obtained cloud worm dataset in order to accurately predict the target class within the data. The detailed explanation can be found in chapter 3, section 3.3.6.8.

2.8 Knowledge discovery in databases for worm detection

Data mining and knowledge discovery were carried out manually in the past. With the passage of time, the amount of such activity had grown and could no longer be processed manually. Moreover, to accomplish something in any related work, discovering essential patterns in data is required. As a result, many software tools were developed to help in discovering hidden data and making assumptions, which after a while became a part of artificial intelligence.

The term Knowledge Discovery in Databases refers to the process of identifying valid, potentially useful, and understandable patterns in the data. KDD process usually consists of five stages which are: selection, pre-processing, transformation, data mining and evaluation. Data mining technique is considered a part of whole KDD

process (Giudici, 2010). For this research, KDD is used as a technique to identify cloud worm patterns in the datasets. All of the KDD processes are summarised in Figure 3.5, Chapter 3 Page 71.

For this research, KDD is used as a technique to identify the EGA patterns within the datasets. This includes datasets preparation, data cleansing, features extraction, clustering, classification, and interpretation. Data mining which is known to be part of KDD is used to extract features of malware detection.

Arshad *et al.*, (2013) used a context-aware method to for intrusion severity analysis using machine learning technique for cloud. In this work, the authors used decision trees supervised technique. Accordingly, a series of rules or questions about the attributes used in decision trees classification technique were used to classify the datasets into classes. For the nature of supervised technique, training and test datasets were used to establish the classifier. Two types of experiments were used; a combination of different datasets and 10 cross validations was conducted.

Bhat *et al.*, (2013) proposed a feature selection technique over the events from virtual machine monitor. This method is based on anomaly but works in parallel to train the system; so, it learns the unknown threats and updates the model accordingly. The combination of NB tree and random forest classifier are used in this model. This work was evaluated on very old dataset which span over a decade.

As for this thesis, KDD and machine learning algorithms have been integrated to optimise the worm detection accuracy in cloud.The subsection demonstrating how each stage of KDD is to be implemented in cloud worm detection is as follows:

2.8.1 Data-preprocessing

Data pre-processing is known as a process for preparing the features before they are implemented on a machine learning algorithm (Kotsiantis *et al.*, 2006). In this research, data pre-processing is to be implemented by collecting the raw cloud worm dataset from virusshare website. Dataset is to be transformed into a new format based on cloud worm's characteristics which could be easily used for the next step in the

analysis. The steps to be involved are: feature selection, data cleansing and data transformation. Detailed explanation can be referred in chapter 3, section 3.3.6.1.

2.8.2 Dynamic Analysis

Dynamic analysis is a process of executing the worm sample, observing the analysis environment, and recording the changes made during the execution time (Kumar *et al.*, 2016). In this research, dynamic analysis is to be used in order to monitor the behaviour of cloud worm to help introduce a new cloud worm classification. The detailed explanation of the analysis is to be found in chapter 3, section 3.3.6.3

2.8.3 Feature Selection

In other similar situations, not all the extracted features may be related to the area of interest especially when dealing with a high-dimensional feature set. Therefore, feature selection allows a more efficient or proficient machine learning procedure by choosing an optimal subset of features (Witten & Frank, 2005). In this research feature selection is to be used to define the characteristics of every cloud worm by using dynamic analysis and also to be tabulated into meaningful dataset for the further analysis. More explanation can be found in chapter 3, section 3.3.6.2.

2.8.4 Data Cleaning and Transformation

Data cleaning is the process of finding out the data that is irrelevant, incomplete, noisy, corrupt or inaccurate from a dataset, and removing the useless data. After the data analysis and cleaning are conducted, the data is then transformed into nominal data with a certain number representation (Abu Zaid, 2013). In this research, data cleaning is also to be implemented in order to remove noise, duplication, and outlier data, and then transform them into nominal data. More detailed explanation can be found in chapter 3, section 3.3.6.4.

2.8.5 Chi- Square and Symmetric Measure

Chi-Square test compares the actual frequencies with the expected frequencies statistically by cross tabulation to verify that the result happens by chance, or not (Greasley, 2007). According to Giudici (2010), the statistical analysis provides added value to the analysis by data mining. In this research, Chi-square and symmetric measure is to be utilised for determining the relationship between cloud worm characteristics, and quantifying the strength of the relationship. The detailed explanation can be found in chapter 3, section 3.3.6.5.

2.8.6 Data Mining

Hand *et al.*, (2001) described data mining as a multi-disciplinary field that contains statistics, artificial intelligence, machine learning, and database technology. The importance of data mining applications is usually assessed to be very high. Many businesses have saved or kept in huge amounts of data over long period of years of operation. Data mining is also able to extract and make use of such valuable knowledge from the data. Data mining are usually connected in finding and structuring large datasets which is also associated with knowledge discovery. In this research, classification and clustering are used in data mining to find hidden patterns. The detailed explanation can be found in chapter 3, section 3.3.6.7

2.8.6.1 Classification

When there is a label connected with each sample, the classification is known as supervised learning. Then, the classifier allocates each sample to the related label based on the sample features. The classifier's job depends on its ability to learn how to differentiate between various samples and their label during the training phase where the classifiers are divided by the labelled sample (Muhammad & Yan, 2015). In this research, classification is to be implemented in order to test the accuracy of the different types of cloud worm assignment. More detailed explanation can be referred in chapter 3, section 3.3.6.8.

2.8.7 Security Metrics

Security metrics is a method that helps to measure, quantify, and classify security based on information. It also defines the threat level that shows how and what level of damage could be done in the system based on Confidentiality, Integrity and Availability (Payne, 2006). In this research, security metrics is to be implemented in order to find out the weight and severity level of cloud worm, and to find the threat level using Confidentiality, Integrity and Availability (CIA). More explanation can be found in section 2.13 and chapter 3, section 3.3.6.6.

2.8.8 Genetic Algorithm

Genetic algorithm (GA) is a search algorithm that is based on the ethics of natural selection and it has been used to solve wide range of problems. Also, GA evolves a population of initial individuals to a population of more qualified individuals, where each individual denotes a solution of the problem to be solved (Dhopte & Chaudhari, 2014; Goranin & Čenys, 2015). In this research, GA is to be used to improve the detection accuracy in cloud worm detection. More detailed explanation can be referred in chapter 3, section 3.3.6.10 and chapter 5.

2.8.9 Data post-Processing

At this stage, the final process of KDD known as data post processing is to complete the pattern extraction from the data that is interpreted for useful knowledge to be produced. More explanation can be referred in chapter 3, section 3.3.6.9.

2.9 Bio Inspired Algorithm for malware detection

Biologically Inspired Algorithms (BIAs) are a process that imitates how organisms solve problems and offer a number of attributes well suited to addressing the challenges presented by future computer networking scenarios. Such future networks require more scalable, adaptive and robust design to address the dynamic changes and

potential failures caused by high heterogeneity and large scale networks (Zheng, 2013).

BIA also consists of numerous sub-categories which were also used in malware detection. Examples of sub-categories include Particle Swam Optimisation (PSO), Negative Selection Algorithm (NSA), Artificial Bee Colony, and Bat algorithm.

There were different uses of these sub-categories under the BIA. According to Sahu and Maharana (2013), NSA under BIA was utilised for explaining the basic characteristics of self-bodies and non-self-bodies. The main aim of this intrusion detection is to identify unauthorised use, misuse, and abuse of computer systems by both internal system users and external fraudsters. NSA was able to monitor any number of hosts in a network by analysing the audit trails of multiple hosts and network traffic. The idea of negative selection algorithm is to come up with a set of detectors and use these detectors for binary classification as self or non-self which has the ability to extract high level knowledge from the generated detectors (Lytvynenko *et al.*, 2015). However, NSA produces poor performance due to scaling issues on real life problems, and this has seriously reduced their usage (Aickelin *et al.*, 2004; Fouladvand *et al.*, 2016). Additionally, there are two limitations to using the NSA algorithm which are: scalability and coverage. These are the main barriers to its success as an effective within the intrusion detection system (Kim *et al.*, 2007). Also, there is no warranty that this algorithm can converge to optimal or close to optimal space coverage with less overlapping issues (Gonzalez *et al.*, 2003).

However, under the BIA sub-categories or characteristics, there are some drawbacks including poor performance and limitation in real world application in detecting malwares. Their applications also usually change and they do not have continuous learning ability in their detection process (Fouladvand *et al.*, 2016).

Also, under the BIA sub-categories, Particle Swam Optimisation (PSO) has been extensively applied to many engineering optimisation areas due to its unique searching mechanism, simple concept, computational efficiency, and easy implementation. It can also search very large spaces of candidate solutions (Nemad & Rane, 2016).

As part of the sub-category of BIA, PSO also tends to have some disadvantages. For instance, it easily suffers from partial optimisation causing it to be less precise in regulating its speed and direction. Hence, the method is unable to work out the problems of non-coordinate system (Aote *et al.*, 2013). It also cannot work out the problems of scattering and optimisation (Bai, 2010). Another limitation is that PSO doesn't guarantee an optimal solution that could be found. Additionally, PSO doesn't utilise the size of the problem being optimised (Nemad & Rane, 2016). Also according to Bai (2010), PSO's theoretical foundation is known to be weak. Hence, it shows that there is no proof mathematically about the convergence and the speed of the convergence. Therefore, the optimist solution related to PSO cannot be ensured in theory. This also demonstrates inadequacy of research regarding the PSO algorithm applications within the related system (Bai, 2010).

Artificial bee colony algorithm (ABC) is known to have predominantly been utilised in the literature as a single objective numerical value optimiser. It has also been used for searching, routing, and assigning the task allocation problems. ABC algorithm can be utilised for solving the multimodal optimisation and multidimensional Problems (Davidovic *et al.*, 2011). Additionally, it can also be used for collective decision making for multi-criteria selection problems (Karaboga & Basturk, 2008; Karboga & Akay, 2009). Therefore, artificial bee colony is not suitable for detection because it does not possess the crossover functionality. This is due to the fact that it has a lot of problems in its diversity method which leads to insufficient accuracy level (Karaboga *et al.*, 2014). Nonetheless, ABC algorithm has some advantages such as fast convergence and high flexibility. However, it has the disadvantages of premature convergence in the later search period and the accuracy of the best value which cannot meet the requirements sometimes (Yan & Li, 2011).

The bat algorithm has been utilised in the existing literature for multi-objective optimisation, constrained optimisation search, combinatorial optimisation and scheduling. Bat algorithm is known to be more useful for simple dimensional problems where convergence often is challenging, same as in the structural design optimisation problems, and the chaotic multi-objective problems (Yang & Gandomi,

2012). The performance of bat algorithm for forced optimisation tasks has been reported to be better for other bio-inspired computing approaches like particle swarm optimisation (Gandomi *et al.*, 2013). In relation to accuracy improvement, it shows that Bat algorithm is also not suitable for accuracy functions due to the fact that its functions relates to convergence and multi objective problems (Gandomi *et al.*, 2013). Another limitation related to bat algorithm is the necessity of the more theoretical background on bat algorithm. Secondly, bat algorithm is inefficient when problems with higher dimensions need to be optimised, and bat algorithm also needs to be used in real world environment (Fister, 2013).

Hence, in this study, the proposed technique known as EGA which is related to Genetic Algorithm is to be implemented in order to improve accuracy in worm detection on a cloud platform (Lu & Traore, 2004).

2.10 The Main Benefits of Genetic Algorithm

Due to the following reasons, this research is to implement Genetic Algorithm (GA). The reason of choosing this algorithm is because it possesses the tendency to solve, search and also optimise problems which is based on the adaptive methods as stated by Vijay and Reddy (2012).

GA has also proven to be beneficial when it is combined with other techniques which can be very useful for detecting malware (Nag & Singh, 2015).

Another reason for choosing Genetic Algorithm (GA) is that it helps in optimising the classification system on malwares. It also helps in predicting malware attacks. This system is based on knowing the targets which are malwares, in this case worms and Trojan horses, in order to understand their behaviour particularly in terms of how they operate (Yusoff & Jantan, 2011). Also, GA has the capability to learn the behaviour of malware (Zolkipli & Jantan, 2010).

Another benefit of using GA can be seen in a research by Mehdi *et al.*, (2009). They used GA to optimise system parameters which are able to detect a malware the first

day they launch their research finding. Additionally, GA is known to give good results in real-time dynamic environments.

Additionally, genetic algorithm is useful and efficient due to the fact that it can rapidly locate good solutions for difficult search spaces. GA is also efficient even when mathematical analysis is not available. It is also useful for complex defined problems because it works with its own internal rules. The few limitations of genetic algorithm include the tendency to change towards local optima instead of the global optimum for the problem when the fitness function is not properly defined (Binitha & Sathya, 2012).

GA also has been proven to show power and potential for a wide range of objective functions. Also, genetic algorithm has been so successful in solving many optimisation problems. Additionally, genetic algorithm has the ability to minimise its learning errors and prediction errors by way of repetitive trails and errors. Another advantage of genetic algorithm over traditional optimisation algorithms is the ability of dealing with complicated problems and parallelism (Yang, 2010).

2.11 Worm detection using genetic algorithm

In 2004, a study was conducted by Lu and Traore based on the detection of malicious attacks through the use of genetic programming. The authors found that genetic programming had a lower false positive rate as well as a higher rate for detection of malicious attacks.

2.11.1 Genetic algorithm definition

According to Vijay and Reddy (2012), soft computing techniques are efficient for the cloud computing security model. Accordingly, genetic algorithm (GA) solves, searches, and optimises problems based on adaptive methods. GA is based on the genetic processes of human's immune system. Survival of the Fittest is the main principles of GA and it always depends on natural selection from many generations of natural populations. Selection, crossover, mutation and fitness functions are involved in the GA process. Kumar and Gohil (2015) stated that GA involves the idea of

natural evolution and it is based on search heuristic. This heuristic is normally used to generate valuable solutions to search and optimise problems. A fitness function measures the fitness by each rule for every rules implementation. Genetic algorithms are used to improve the accuracy of IDS by selecting the network features and determining the optimal parameters to improve optimisation (Modi *et al.*, 2013). In conjunction with this thesis, a new technique to detect and respond to worm attacks in cloud computing environment by using genetic algorithm has been designed and evaluated. The details and experimental results can be referred in Chapter 5.

2.11.2 GA Approach for Cloud Worm Detection

Genetic algorithm (GA) is a search algorithm based on the principles of natural selection and it has been deployed to solve a wide range of problems. GA evolves a population of initial individuals to a population of high quality individuals, where each individual represents a solution of the problem to be solved (Dhopte & Chaudhari, 2014; Goranin & Čenys, 2015). Each individual is called chromosome, and is composed of a predetermined number of genes. The quality of each rule is measured by a fitness function as the quantitative representation of each rule's adaptation to a certain environment. The procedure starts from an initial population of randomly generated individuals. Then, the population is evolved for a number of generations while gradually improving the qualities of the individuals in the sense of increasing the fitness value as the measure of quality. During each generation, three basic genetic operators are sequentially applied to each individual with certain probabilities; these are: selection, crossover, and mutation.

2.11.3 Parent Selection Process

Reproduction (or selection) is an operator that makes more copies of better strings in a new population. Reproduction is usually the first operator applied on a population. Reproduction selects good strings in a population and forms a mating pool. This is one of the reasons for the reproduction operation to be sometimes known as the selection operator. Thus, in reproduction operation, the process of natural selection causes those individuals that encode successful structures to produce copies more frequently (Ahmad izar *et al.*, 2015).

2.11.4 Crossover

A crossover operator is used to recombine two strings to get a better string. In crossover operation, recombination process creates different individuals in the successive generations by combining material from two individuals of the previous generation. In reproduction, good strings in a population are probabilistically assigned a larger number of copies, and a mating pool is formed. It is important to note that no new strings are formed in the reproduction phase. In the crossover operator, new strings are created by exchanging information among strings of the mating pool (Jin *et al.*, 2015).

2.11.5 Mutation

Mutation adds new information in a random way to the genetic search process. It is an operator that introduces diversity in the population whenever the population tends to become homogeneous due to repeated use of reproduction and crossover operators. Mutation may cause the chromosomes of individuals to be different from those of their parent individuals. Mutation in a way is the process of randomly disturbing genetic information. They operate at the bit level; when the bits are being copied from the current string to the new string, there is a probability that each bit may become mutated. The purpose of mutation is to create a point in the neighbourhood of the current point, thereby achieving a local search around the current solution. The mutation is also used to maintain diversity in the population (Abdi *et al.*, 2014).

2.11.6 Utilisation of GA for worm or malware detection

In recent times, various studies used genetic algorithm for malware detection in cloud (Kumar & Gohil, 2015; Majeed & Kumar 2014; Li *et al.*, 2012; Goyal & Aggarwal, 2012). According to Yusoff and Jantan (2011), GA helps to optimise the classification system for malwares, and at the same time, helps to predict malware attacks. This system is based on understanding the target which is a malware, in this case worms and Trojan horses, to understand the behaviour of its operations. Yusoff and Jantan (2011) suggested the combination of Genetic Algorithm (GA) along with Decision Tree (DT) since it is a more compatible algorithm to study the behaviour. Some of the

key elements in this framework include the sample of malwares, the virtual environment, knowledge storage, classifier, as well as the class operation.

The study by Majeed and Kumar (2014) reported that it is important to focus on the area of intrusion detection methods along with genetic algorithms. Some of the key areas of GA are focused on optimisation, design of automatic models, as well as classifications. A study by Chittur (2001) also concluded that the use of GA is necessary to optimise the main purposes within the detection system. It was also reported that there are some key differences in the GA system as opposed to the conventional system. According to Majeed and Kumar (2014), the GA system works based on the codes that are prevalent in the scope of the identified problems, compared to the traditional systems that work on the problem scopes itself. Besides that, GA systems provide a higher number of effective solutions compared to the conventional method of producing only one method to solve a problem. Therefore, the GA system is more efficient. Conventional systems employ the use of derivatives to evaluate a solution, whereby the GA system employs the use of a fitness function to evaluate the highly optimal solution that it has produced. The study also reported that GAs use probabilistic operators for transitions compared to conventional systems that use of deterministic operators (Majeed & Kumar, 2014). Based on the above existing works, the importance and flexibility of the GA system in detecting intrusions in a cloud environment have been highlighted.

Another study conducted by Gupta and Shinde (2011) added a new system that used the certain procedures that helped to screen the classifier's decision, which then reduces false-positive rate. The study also used the feature extraction technique with the genetic algorithm technique to lower the amount of files that the system processes. The study also found that genetic algorithm has more advantages in detecting intrusions because it is able to look at many directions in a short duration. This is an important consideration due to the fact that there are large numbers of data to be detected because of the high number of users in a cloud environment.

GA is able to eliminate unsuccessful means and pursue other options, which maximises the chances of detecting intrusions in the system. This allows easier flow of instructions to the system, which makes is more efficient and fast. Another

advantage of the GA system is that it can easily be modified according to newer discoveries found in the intrusion detection area of study, making GA one of the most flexible and open systems to work with. Moreover, the adaptability of the GA system is a positive trait especially due to the emerging new attacks (Bankovic *et al.*, 2009).

GA is used to select network features (to determine optimal parameters) which can be used in other techniques for achieving result's optimisation and improving the accuracy of IDS (Dhanalakshmi & Ramesh Babu, 2008; Li, 2004). Further, Gong *et al.* (2005) used seven features (Duration, Protocol, Source_port, Destination_port, Source_IP, Destination_IP, Attack_name) of captured packet. They used support confidence based framework for fitness function, which is simple and flexible. Generated rules are used to detect network intrusions. The paper uses quantitative as well as categorical features of network for generating classification rules. This increases the detection rate and improves accuracy. However, limitation of this approach is the best fit problem. Meanwhile, Lu and Traore (2004) presented a GA based approach to generate rules from network features. They used support confidence-based fitness function for deriving rules, which classifies network intrusions effectively. However, the training period for the fitness function takes more time. On the other hand, Xia *et al.*, (2005) proposed information theory and GA based approach that is used to detect abnormal behaviour. It identifies small number of network features closely with network attacks based on mutual information between network features and type of intrusion. However, this approach only considers discrete features. Additionally, Dhanalakshmi and Ramesh Babu (2008) proposed a method which is used to detect misuse and anomaly by combining fuzzy and GA. Fuzzy is used to include quantitative parameters in intrusion detection, whereas GA is used to find best fit parameters of introduced numerical fuzzy function. This approach solves best fit problem as reported by Lu and Traore (2004). In cloud environment, selection of optimal parameters (network features) for intrusion detection will increase the accuracy of underlying Intrusion Detection System (IDS). For that, Genetic algorithm (GA) based IDS can be used in Cloud.

OlexGA was proposed by Pietramala *et al.*, (2008), and it has been applied in machine learning techniques. The Olex's hypothesis language consists of rules with one positive conjunction of terms and (zero or) more negative ones. Thus, Olex

predictions require testing the simultaneous presence of several terms (forming the positive conjunction) along with the simultaneous absence of several sets of terms (each forming a negative conjunction). New parameters are proposed in GA for cloud worm detection by OlexGA implementation. OlexGA algorithm has the following parameters as presented in Table 2.5. OlexGA has been used as benchmark to evaluate the experimental results of this thesis.

Table 2.3: OlexGA parameters

GA Parameters	**Description**	**Default Value**
Xover	Type of cross-over.	Uniform Xover
Xover Rate	Probability of two individuals (chosen by the selection algorithm) of undergoing reproduction.	1.0
Class Index	Position of the label of the category being learned in the class attribute list - 0 if the category is in the first position, 1 otherwise.	0
Elitism Rate	Percentage of current best individuals that are passed to the next generation.	0.2
mutation Rate	Chance of a gene of being flipped by standard mutation.	0.001
Num of features	Number of both positive and negative candidate features. Suggested values range between 30 and 100.	*50*
Num Of Generations	Number of new populations created by the GA.	200
Num Of Runs	Number of times the genetic algorithm is run (keeping the same input parameters). The best classifier is returned.	1
Population Size	Number of individuals in the population.	*500*
Scoring function	Type of feature selection function used to select candidate features.	Chi Square
Selection Algorithm	Type of algorithm used for the selection of the parent individuals for reproduction.	Tournament Selection

According to a write-up by Rahate and Lobo (2013) and Khedikar and Kulkarni (2014), OlexGA was used for developing classification. It was used together with Hadoop technology which had components that were used to store large amount of data and could also carry out computations at a faster level. They also explained that OlexGA is known as a plug-in used for implementing a parallel GA within the data mining package. It was also able to give efficient text categorisation.

OlexGA was also utilised for diagnosis of health processes in the medical field. The OlexGA was used to detect different diseases in specific stages which also improved the adaptive nature of new symptoms through its classification processes in order to increase an accuracy of prediction of diseases. With the help of OlexGA, the adaptive health system was enhanced automatically to adapt itself to new symptoms and different tests in the patients (Shivagude & Kulkarni, 2016).

According to Manjula (2011), OlexGA has proven to be one of the most highly competitive among the top-performing learning algorithms utilised for text categorisation such as C4.5, SVM and Naïve Bayes. It also turned out that OlexGA tends to have a more efficient induction method which is faster than the Ripper and C4.5. OlexGA's effectiveness on rule induction is also known to be a result of its innate ability to catch up with interaction of attributes. In the experiment explained in Manjula's (2011) write-up, it has proven that OlexGA can bring classifiers known to be compact and accurate. It should be noted that accuracy is observed as a consequence related to a powerful hypothesis language. On the other hand, efficiency occurs as a result of effective optimisation. As a result of the experiment performed by the publisher above, it has also proven that performance can be further improved by way of fine-tuning of the GA parameters.

One of the reasons for choosing OlexGA as a rule based system is its design which is able to classify texts. It also has rules that consist of one positive conjunction and negative conjunction. OlexGA was also useful on a research for GAMON in which it was used to access the effectiveness of its extension. GAMON is known to be a task-specific Genetic Algorithm (GA) utilised for exploiting the lattice-based structure of the hypothesis space, efficiently and also for improving accurate hypotheses. By implementing OlexGA as one of its algorithms, it helped the Gamon project to achieve efficient outcome by making it provide high predictive accuracy among a wide class or problem domains. It also aided in constructing an easy and compact model in which it facilitated human comprehension (Rullo *et al.*, 2012).

According to Grasso *et al.*, (2009), another Domain in which OlexGA is applied is e-Government. The system was legally built according to authorised acts and decrees permitted by public authorities which are classified. The system uses a strategy that is based on an archive that is known to contain a full list of arranged words into synonyms with concepts related to its juridical expressions used by the Italian Parliament. The performance of OlexGA was very efficient by which it obtained an f-measure of 92% with a mean precision of 96% in relation to real-world documents. The authors above also stated that OlexGA was also implemented on e-medicine domain. It was used for building up a system that could be able to classify case document and histories automatically that contained clinical diagnoses.

In other Articles written by Stephanie Chua and Coenen (2013), OlexGA was also used to generate rules by using genetic algorithm. However, the generated rule still utilised the same template of a positive feature including a zero negative feature. OlexGA system still performed at a more efficient level than other techniques such as C4.5, SUM, Ripper and NB.

According to an earlier write-up by Rullo *et al.*, (2009), OlexGA has achieved high performance for learning the rule-based text classifiers from training sets that make classifiers which are comprehensible and compact. In comparing the analysis carried out by some advanced learning methods such as Ripper, Naïve Bayes (NB), C4.5, Linear Logistic Regression (LLR) and SMV, it was proven that OlexGA has more competitive advantage in the instances and accuracy. These experiments were carried out on three data collectors which were called OHSUMED, REUTERS-21578, and ODP. The experiment shows that OlexGA makes use of numerous acceptable properties as follows: It creates classifiers that are known to be compact and comprehensible for relatively smaller categories, it tends to be accurate which also shows that it is not inclined to only majority classes, and lastly, it tends to be robust by displaying the same behaviour on the datasets it experimented on. Table 2.4 as shown below, summarised Genetic algorithm, OlexGA and other Bio inspired algorithms for malware detection.

Table 2.4: Summarisation of GA and other Bio inspired Algorithms

Authors	Name of Algorithm	Strength	Weakness
Nemad and Rane (2016), Aote *et al.*, (2013) and Bai (2010)	PSO	• Simplicity and easy implementation • It can search very large spaces of candidate solutions	• Easily suffering from partial optimisation. • The method is unable to work out the problems of non-coordinate system. • It cannot work out the problems of scattering and optimization. • It doesn't utilise the size of the problem being optimised. • PSO theoretical foundation is known to be weak. • It shows that there is no proof mathematically about the convergence and the speed of the convergence. • Doesn't guarantee the discovery of optimal solution. • It shows that an optimist solution related to PSO cannot be ensured in theory. • It is a minimum research on the PSO algorithm applications within related system.
Fouladvand *et al.*, (2016), Lytvynenko *et al.*, (2015), Dixon (2010), Kim *et al.*, (2007), Aickelin *et al.*, (2004) and Gonzalez *et al.*, (2003)	NSA	• Larger area of non self space is covered by fewer detectors. • Extracts high level knowledge from the generated detectors.	• Does not have continuous learning ability in its detection process. • Produces poor performance due to scaling issues on real life problems and this has reduced its use. • It has limitation related to scalability and coverage and these are the main barriers to its success as an effective within the intrusion detection system. • There is no warranty that this algorithm can converge to optimal space coverage.
Karaboga *et al.*, (2014), Davidovic *et al.*, (2011), Yan and Li (2011), Karboga and Akay (2009), and Karaboga and Basturk (2008).	Artificial Bee colony	• Can be used for solving multidimensional and multimodal optimisation problems • Fast convergence and high flexibility	• Premature convergence in the later search period and the accuracy of the optimal value which cannot meet the requirements sometimes. • Itis not suitable for detection because it does not possess the crossover. • It has a lot of problems in its diversity method and also leads to insufficient accuracy level.

Fister (2013), Gandomi *et al.*, (2013), Yang and Gandomi (2012), and Yang (2010).	Bat Algorithm	• It is useful for simple dimensional problems. • Simulating annealing can almost guarantee the discovery of the optimal solution.	• It is not suitable for accuracy functions due to the fact that its functions relate to convergence and multi objective problems • More theoretical background on bat algorithm is needed. • Adjustment in parameters does affect the convergence rate of the optimisation process. • Ineffictent when problems with higher dimensions need to be optimised. • Bat algorithm needs to be used in real world environment.
Nag and Singh (2015), Majeed and Kumar (2014), Modi *et al.*, (2013), Binitha and Sathya (2012), Vijay and Reddy (2012), Yusoff and Jantan (2011), Yang (2010), Zolkipli and Jantan (2010), Mehdi *et al.*, (2009), and Lu and Traore (2004).	GA	• Tendency to solve, search and optimise problems. • Can be very useful for detecting malware. • Helps in optimising the classification system on malwares. • Helps in predicting malware attacks. • Has the capability to learn the behaviour of malware. • Ability to optimise system parameters which are able to detect a malware. • Genetic programming had reported a lower false positive rate as well as a higher rate for detection of malicious attacks. • Gives good results in real time dynamic environments. • Provides a higher number of effective solutions compared to other methods. • Useful and efficient; ability to rapidly locate good solutions for difficult search spaces. • Useful for complex defined problems. • Shows potential and power for a wide range of objective functions. • Ability to minimise its learning errors and prediction errors by way of repetitive trails and errors.	• Possible tendency to change towards local optima instead of the global optimum for the problem when the fitness function is not properly defined. • Can provide the same amount of computation time for specific optimisation problems.
Shivagunde and Kulkarni (2016), Khedikar and Kulkarni (2014), Lobo and Rahate (2013), Manjula (2011), Rullo *et al.*, (2012), Grasso *et al.*, (2009), Chua and Coenen (2013), and Rullo *et al.*, (2009).	OlexGA	• Can bring classifiers known to be compact and accurate. • Implemented in different fields. • Assists in the achievement of efficient outcome by providing high predictive accuracy among problem domains. • Ability to detect different diseases in specific stages. • Ability to increase the accuracy of prediction of diseases. • Improves the adaptive nature of new symptoms through its classification processes.	• Performance can be further improved by way of fine-tuning of the GA parameters. • Loss of diversity • Smaller chance to select weak Individual. • Gives bias with unlimited spread. • It can give new child new characteristics quite similar to the parent characteristics. • Generate an offspring that is a little bit different from their parents.

		• Fosters efficiency through effective optimisation. • Ability to give efficient text categorisation. • Applied in different fields such as health care, text classification, e-government, and e-medicine. • Ability to store large amount of data. • Ability to carry out computations at a faster level. • Among the most competitive among the top performing learning algorithms. • Tendency to have a more efficient induction method.	

The table above shows the summary of different algorithms that have been applied by numerous researchers, alongside their strength and weakness. In this study, further research is to be carried out in order to come up with improvement, which is, to consist of more efficient and effective process in worm detection. Based on all the existing works stated above, it is clearly seen that GA could be further improved and has promising result in improving the accuracy of cloud worm detection. As also seen in the table above, GA tends to have more strength as compared to other algorithms. For instance, part of the GA's advantages is that it has the capability to learn the malware behaviour. It also helps in predicting malware attacks. It is also noted that GA have reported lower false positive rate as well as higher rate for detecting malware attacks. Therefore, this research is focused on genetic algorithm to find better solution to improve cloud worm's accuracy detection and prevention.

On the other hand, based on previous research as stated above, OlexGA was implemented in different fields. It is observed that OlexGA proved to be more efficient and effective in different fields. It is also noted that OlexGA tends to be more stable and powerful. Its induction method is also more efficient and is more accurate as compared to other algorithms. However, OlexGA have the potential to be utilised for malware detection domain, but it has never actually been implemented or used for malware detection.

Hence, OlexGA suffers from the use of the tournament selection which leads to smaller chances of selecting weak individuals which leads to loose individual

diversity. Additionally, it also suffers from uniform crossover that creates offspring which is slightly different from their parents. In relation to mutation, it uses substitution technique which gives it new characteristics that are quite similar to the parent characteristics. Another limitation is related to the lack in evolution technique. Therefore, this research aims to handle the improvement of OlexGA by re-implementing OlexGA itself in order to enhance and also increase the accuracy in detecting worm within cloud environment.

Bhat *et al.*, (2013) proposed an Anomaly Intrusion Detection System using the machine learning approach for virtual machines on cloud computing. In this regard, the authors proposed the application of feature selection over events from Virtual Machine Monitor to detect anomaly in parallel to training the system so it will learn new threats and update the model. The experiment has been carried out on NSL-KDD'99 datasets using Naïve Bayes Tree (NB Tree) Classifier and the hybrid approach of NB Tree and Random Forest.

Based on all the existing works stated above, it is clearly seen that GA could be further improved and has promising result to improve the accuracy detection. Therefore, this research is focused on genetic algorithm to find better solution to improve cloud worm's accuracy detection and prevention.

2.12 Existing cloud worm's detection technique

A malware detection technique to detect malware in cloud computing had been presented by Thu *et al.*, (2015). This technique combines system call monitoring and system call hashing. This combination is used in addition to support vector machine based external monitoring on the host. During system call monitoring process, all system calls triggered by the users are monitored through the parameter before execution. Meanwhile in the system call hashing, the technique checks all the stored copies of monitored system called before installation. Then, the support vector machine is used to classify all the malware attacks based on guest behaviour in virtualised cloud system. However, some malwares are also able to slip through the classification process due to lack in correct detection in the process. This shows that

there is no accuracy level. By improving the accuracy level, the malwares detection can also be improved, therefore making the call monitoring process much safer and efficient. Hence, there is a need for further enhancement to increase the accuracy rate of attacks detection.

Watson *et al.*, (2015) proposed a technique based on support vector machine to detect malware and DOS attacks in cloud computing infrastructure. This technique has the ability to monitor system behaviour to detect malware at hypervisor level in cloud computing system through the utilisation of features collected at the system and network levels of a cloud. However, there are several types of threats that can be found in cloud virtualisation. Hence, absence of accuracy level in the system can cause malware intrusion. Therefore, the accuracy level needs further improvement to make sure malware threats are detected correctly.

Marnerides *et al.*, (2013) presented anomaly detection technique to detect malware for virtualised cloud environment. This technique uses both network analysis and system analysis engines to analyse, monitor and detect malware activities in virtual machines by monitoring the processes and usage of memory. However, low accuracy in this process might enable malwares to invade cloud environment.

Hatem *et al.*, (2014) proposed a new malware detection technique in cloud computing environment. In this technique, dynamic and static signature detections are used to identify malicious and unwanted software in cloud which provides an enhanced operation for the detection of malware in cloud while also improving the forensics capability. However, the process might be lacking in accuracy. Hence, this technique needs more improvement in malware detection particularly in terms of its accuracy level to provide an enhanced functionality of detection while providing correct results of detection.

Martínez *et al.*, (2010) proposed an ontology-based malware detection technique called uCLAVS (University of Caldas' AntiVirus Service) for cloud. This technique combines ontology with intrusion detection to represent the signatures for known and novel attacks. Additionally, the technique uses a K-mean clustering technique to classify the malware attacks into different categories. Based on these categories,

malware detection is performed to detect malicious software. In cloud, users have the ability to access various applications through internet browser, which can encourage any malicious software to be inserted into user's machine. Thus, a malware detection using these definitions is performed to detect and reduce threats in those applications. Due to these incidences, the accuracy level might be degraded or become low. However, a better accuracy level is required to maintain the precise detection of malwares.

Silakari and Chourasia (2016) proposed a technique to detect malware in cloud infrastructure using Accelerated Chaotic Map Particle Swarm Optimisation (ACMPSO-k means). This technique detects three types of attacks: DOS, spam, and phishing attacks, in cloud computing infrastructure. The technique combines both PSO and K-Means methods in order to optimise and provide an enhancement for detection process, and to get accurate solution for the problem. K-means are search-based optimisations which are used to classify data and provide optimised solution for the problem. However, the results of this study showed that the average accuracy detection rate is relatively good, but with further augmentation, it could lead to increase in the accuracy rate of detection.

Bhat *et al.*, (2013) proposed an anomaly detection system using machine learning technique for virtual machines in cloud computing. This technique works based on feature selection over the events from Virtual Machine Monitor to observe the activities of the virtual machine. In particular, the technique detects the anomalies in the virtual machine. In addition, the technique trains the system so that it can learn unknown threats and accordingly update the model. The experiment has been carried out on NSL-KDD'99 very old datasets using Naïve Bayes Tree (NB Tree) Classifier, and hybrid approach of NB Tree and Random Forest. Then again, in the absences of improving accuracy as one of the training functions, this may result in malware invasion.

Another intrusion detection system was proposed by Dhage and Meshram, (2012) which consists of finding out malicious attacks within cloud system by using signature method which also protects against the blocking features. The system focuses on monitoring and managing numerous volumes of traffic within the system. The main

aim is to locate a specific signature known to be threats which will be checked and monitored against known signatures that have been formed on malicious attacks in order to find a match. However, this model was not implemented in real or simulated environment. The proposed approach is unable to detect unknown attack and also suffers from the lack of accuracy. In other instances, it may include scalability and sensitivity of the system.

Zhang *et al.*, (2014) proposed a malware detection technique for cloud computing. The technique is similar to cloud antivirus technique. Multiple engines are used to detect the malware. These engines include Threat expert, CW sandbox, Anubis, Joe sandbox and Cuckoo sandbox. The authors suggest combining these engines in order to improve the detection rate. However, the results for the combined detection rate are not presented in the paper (Alam *et al.*, 2014).

On the other hand, Qaisar and Khawaja (2012) reported that a counter measure in solving worm attacks in cloud is by checking the authenticity of the messages received, and then keeping a record of the image file that was requested. This should be done to allow the comparison of this image file against the hash that is used in future services using hash functions as well as other requests. Then again, attacker can create a legitimate hash value to deal with or enter cloud system.

One of the proposed solutions was by making sure cloud providers are able to create an image user (VM) which will be stored in the image repository system of cloud. Another recommendation was the application of higher integrity by cloud service providers because it will impede the attackers or intruders from invading cloud system. Qaisar and Khawaja (2012) also suggested that in utilising the hypervisor process for scheduling instances, checks on reliability are done by taking from FAT table usually located in users' VM. Another option given is to keep or save the users' version of OS that is normally performed as users open an account through cloud provider giving them the ability to verify the OS type with new application before initiating an instance in the cloud (Zunnurhain & Vbrsky, 2010).

Various methods such as, neural network, support vector machine, and genetic algorithm were used to detect worm or malware attack in cloud computing. For instance, Aljurayban and Emam (2015) presented Layered Intrusion Detection

Framework (LIDF) which can be applied on the various layers of cloud computing to identify the presence of normal traffic from the monitored cloud traffic. The proposed framework is based on data mining technique specifically Artificial Neural Network for detection activity. A layer IDS model is also proposed by this work to handle incoming traffic in scalable large network traffic, and control administrative application and data in cloud. This cloud IDS handles large flow of network traffic, analyses them, and generates efficient reports by integrating the knowledge of behaviour analysis to identify and detect intrusions at their earlier stages. The framework displayed an effective outcome in the process of the experiment which was able to reduce false positive rate but also showed a weakness in identifying true positive rate in detection process. Additionally, this shows a need for more enhancements including the accuracy level within different cloud structures (Heenan & Moradpoor, 2016).

Table 2.5: Existing cloud worm detection technique.

itle	**Detection Methods**	**Advantage**	**Weakness**	**Parameters**
Detection malware and ernel level root kits in loud computing (Thu *et l.*, 2015)	Using system call monitoring and system call hashing with support vector machine based external host monitoring system in the guest kernel	It provides enhanced functionality to provide security for virtualized cloud system	-This technique suffers from the accuracy of malware detection in cloud and needs further improvement to increase the accuracy of attack detection in cloud environment (Nancy,2016)	Not addressed
Malware Detection in loud Computing nfrastructures (Watson *et l.*, 2015)	Using support vector machine based malware detection technique	It has ability to provide classification for the neuromas malware attacks by support vector machine learning algorithm	-It is not able to detect various threats in virtualization cloud (VM). -it needs further enhancement mechanisms to increase the accuracy results for malware detection in cloud.	Accuracy, Precision, Recall,Fscore, G mean
Malware Analysis in Cloud omputing: Network and ystem Characteristics Marnerides *et al.*, 2013)	Malware detection in virtualise cloud environment	Effective for detecting Kelihos injection.	-Insufficient in detecting other malwares in cloud	Not addressed

Malware Detection in Cloud Computing (Hatem *et al.*, 2014)	Using dynamic and static signature detection for malware detection in cloud	It is flexible and adds some enhanced operation for detection of malware in cloud	-It needs more improvement for malware detection to increase the accuracy rate of detection	Rate of detection only -Does not provid parameters
Malware Detection based on Cloud Computing integrating Intrusion Ontology representation (Martinez *et al.*, 2010)	Using ontology combined with intrusion detection to detect known attacks and using K-mean clustering to classify malware into different classes	Provides more information on known malware attacks	-Encourages malicious software to be inserted into user's machine due to usage of other applications. -Incresed rate of detection is needed	-Rate of detection only - Does not provid parameters
Intrusion detection system in cloud computing environment (Dhage & Meshram, 2012)	Signature based method to detect malicious attacks in cloud	Selection according to the signatures that are classified as threats. These items are then further monitored and checked against known signatures	-Uunable to detect unknown attack and also suffers from the lack of accuracy, scalability and sensitivity of the system.	Not addressed
Machine Learning Approach for IntrusionDetection on Cloud Virtual Machines (Bhat *et al.*, 2013)	Observation of the activities of the virtual machine to detect the anomalies in the virtual machine by using the hybrid approach of NB Tree and Random Forest classifier.	Good performance, good accuracy, and low false positive rate for NB tree/random forest hybrid implementation Good performance for monitoring virtual machineactivities	-Work evaluated using very old dataset -High false positive rate for non-hybrid implementations	Accuracy, Precision, Recall, F-Measure
Security Attacks and Solutions in Clouds (Zunnurhain &Vbrsky, 2010)	-Using FAT technique. -Saving the records on user's OS types.	- It increase the usage of integrity by cloud service providers	-Untested theoretical approach without tested	Not addressed
MalwareDetection Techniques in Cloud Computing Infrastructure using ACMPSO-k means (Silakari& Chourasia, 2016)	Using the combination of both PSO and K-Means method to enhance the malware detection process and classify dataset to discover the malicious attacks	It has the ability to provide classification for the, Dos, spamming and phishing attacks	-insufficient to detect other malwares in cloud -Not accurate results	Accuracy, Precision, Recall, F score ,G mean
Framework for Cloud Intrusion Detection System Service (Aljurayban & Emam,2015)	Using Artificial Neural Network for monitoring the incoming traffic and control application and data in cloud	Provides an optimised solution for handling intrusions at numerous layers of the Cloud network infrastructure.	Weakness in identifying true positive rate in the detection process including low accuracy	Precision Accuracy F-measure

Cloud Computing: Network /Security Threats and Counter Measures Qaisar & Khawaja, 2012)	-Identifying the legitimacy of messages received in cloud -Storing the original image file that was requested by using hash function -Store the original image file that was requested by using hash function	Good to check the authenticity of the messages received in cloud to give more information to cloud provider	-Theoretical approach without tested	Not addressed

In this research, genetic algorithm is used to enhance cloud worm detection and classification by using different method of selection, Crossover, and mutation. Furthermore, the sustainable nature of generation production and best parents were also selected before the creation of new generation. The technique aims to enhance True Positive, False Positive, Precision, Recall, F-Measure, and accuracy rate.

2.13 Security Metrics

Security metrics assists in identifying what metrics are by depicting a differentiation between metrics and measurements. Measurements provide single point in time views of specific and discrete factors, while metrics are resultant by putting together a predetermined reference point. Metrics are generated from analysis. Metrics may be objective or subjective human interpretations of data and measurements, and are also objective raw data. Good metrics must have SMART capabilities, i.e. specific, measurable, attainable, repeatable, and time-dependent (Payne, 2006).

Metrics can be utilised as an effective tool for security managers to detect the effectiveness of different components of their security programs, the security of a particular system, product or process, and the capability of staff or departments within an organisation to address security issues for which they are responsible. Metrics can also help identify the level of risk if the proper action is not taken, and in that way provide guidance in prioritising corrective actions. Additionally, they may be used to increase the level of security awareness within the organisation. Many research works focused on security metric to measure the threats on ICT in general (e.g., Savola, 2007; Savola, 2008; Jafari *et al.*, 2010).

2.13.1 Weight and Severity

Weight is measured based on the security level of five main features which are infection, activation, payload, propagation and operating algorithm. Based on CIA (Confidentiality, Integrity and Availability), weight value is defined. Based on the weight value, severity value is also defined.

2.13.2 Confidentiality, Integrity and Availability (CIA)

CIA is known to be a model proposed to direct policies related to information security in any organisation. The model can also be referred as AIC Triad (availability, integrity and confidentiality) to prevent confusion with the Central Intelligence Agency. There are three elements of the triad which are the most important components of security. In this context, confidentiality (secrecy) is a set of rules that limits access to information while integrity is the guarantee that the information is trustworthy and accurate, and availability is also a guarantee of reliable access to the information by authorised people.

According to Swanson, (2001) from National Institute of Standard and Technology (NIST), CIA is known to consist of the following components:

i. Confidentiality - The information requires full protection and prevention from unauthorised disclosure.

ii. Integrity - The information must be protected from unauthorised, unanticipated, or unintentional modification, alteration or changes. This includes, but is not limited to:

   a. Authenticity –A third party must be able to confirm that the content of a message has not been changed during transit.

   b. Non-repudiation – The origin or the reception of a specific message must be verifiable by a third party.

   c. Accountability - A security goal that generates the requirement for actions of an entity to be traced uniquely to that entity.

iii. Availability- The information technology resource (system or data) must be available on a timely basis to meet demands or to avoid considerable

losses. Availability also includes making sure that resources are used only for needed purposes.

In addition, all the above components can be grouped by the sensitivity or threat level with high, medium and low. Federal Information Security Management Act of 2002 (FISMA) (NIST, 2004) describes three levels of potential effects on organisations or individuals if there should be a breach of security:

i. The potential impact would be considered LOW if the loss of confidentiality, integrity, or availability could be expected to have some degree of adverse effect on organisational operations, organisational assets, or individuals.
ii. The potential impact is MODERATE if the loss of confidentiality, integrity, or availability could be expected to have a severe adverse effect on organisational operations, organisational assets, or individuals.
iii. The potential impact is HIGH if the loss of confidentiality, integrity, or availability could be expected to have a serious or catastrophic effect on organisational operations, organisational assets, or individuals.

2.14 Cloud Worm Response

Most studies in incident response in cloud computing field are commonly theoretical and only present ideal model for the response (Patrascu & Patriciu, 2013). A typical standard model for incident response was reported by NIST (Kent *et al.*, 2006) which is the most remarkable study in this research area. In the field of computer forensics, four basic steps are suggested: collection, examine analysis and reporting. But, these steps help to detect specious behaviour of worm or malware. Thus incident response is a great research issue and it is really challenging to suggest for analysis and isolation after detection.

Researchers such as Grobauer and Schreck (2010) and Chen *et al.*, (2012) presented possible challenges on incident response and handling incident in cloud computing environment, and also thought of a well-defined efficient process to be reflected in cloud incident handling. They also suggested about the changes to be made during

migration into cloud by considering the current process. These recommendations are made by bearing in mind the perception of a security manager and client.

Other researchers like Priya and Prabakaran (2012) stated that one of the central purposes in cloud security is to find the existing vulnerabilities and security holes within cloud environment which is the same as other computer technological fields. In order to achieve usage of available resources, scalability, efficiency and flexibility, cloud providers have to face major challenges. On the other hand, they have to ensure security to their corporate and individual client. Thus, response to cloud worm detection is very important (Rizvi & Mitchell, 2015).

Patrascu and Patriciu (2013) suggested that incident responders and forensic investigators must now rely more on live collection and analysis of system RAM. They also presented a new way of monitoring user activity in cloud environments using a secure cloud forensic framework. For important high end cases, live response has been utilised for six to eight years for important high end cases.

This research uses GA for worm detection and suggests how the system responds to cloud worm detection. Response is an important part to save user from negative effect and it also prevents financial loss in order to reduce damages. Once worm threat is detected, it should be able to isolate and send note to the administrator for shutdown.

2.15 Performance Evaluation Criteria

Vaishnaw and Tandan (2015) used evaluation criteria using parameters to find out the result of the accuracy performance in their study.

Accordingly, this research is to perform the analysis for the worm dataset that has been obtained from virushare.com. The analysis is also to be carried out which consists of an evaluation criteria based on the chosen parameters which are to be used for measuring the accuracy of the detection rate. The parameters to be used are listed below:

True Positive (TP): relates to the number of positive examples correctly predicted by the classification model. In this research, TP refers to the number of the samples that are correctly classified and identified as malicious.

False Negative (FN): relates to the number ofpositive examples wrongly predicted as negative by the classification model. In this research, FN refers to the number of the samples that are incorrectly classified and identified as non malicious.

False Positive (FP): relates to the number ofnegative examples wrongly predicted as positive by the classification model. In this research, FP refers to the number of the samples that are incorrectly classified as malicious.

True Negative (TN): relates to the number of negative examples correctly predicted by the classification model. In this research, TN refers to the number of the samples that are correctly classified and identified as non malicious.

Accuracy: accuracy of the classifier is the proportion of instances which are correctly classified. In this research, accuracy refers to the number of the samples that are correctly classified.

F-measure: the value that assesses the entire system's performance by combining precision and recall into a single number.

Precision: precision is the level of strength of the classifier in terms of predicting the positive instances.

Error rate: error rate calculated by subtracting accuracy from 1 which represents the error of the classifications (Aung & Zaw., 2013; Vaishnaw & Tandan., 2015; Narudin *et al.*, 2016).

This research evaluates TP, FP, Precision, Recall, F-Measure and accuracy rate. The detailed usage and calculation of these parameters can be found in chapter 3, section 3.3.8.

2.16 Summary

In conclusion, chapter two discusses some of the key definitions and concepts that are used in this research. First of all, the concept of cloud computing was discussed. Here, the various definitions were discussed, along with some of the key characteristics of cloud computing. Five main characteristics were identified which show that cloud computing uses shared resources. Furthermore, cloud computing has large volume of users and high flexibility, and also allows users to only pay for resources they have used from the thousands provided as well as the opportunity to self-provision resource. Then, the different cloud service models were discussed, followed by the four main deployment models which are: public, private, hybrid and community cloud. Following this, the vulnerabilities of cloud computing was analysed according to the study by Hashizume *et al.*, (2013). More importantly, the treats in cloud computing and some of its key challenges were discussed. This was important in order to identify the key problems faced by cloud users and providers.

Next is the comparison of worm attacks with other malicious attacks in cloud which was discussed. Additionally, some of the problems confronted by cloud providers and some of the methods that were presently used in the cloud environment were highlighted. This is followed by the discussion of worm attacks in cloud environment, alongside the analysis of the worm detection techniques in cloud. Next, worm detection using genetic algorithm was discussed and related studies which used genetic algorithm for malware detection were also highlighted.

Here, many studies were discussed, and followed by a brief summary of the key findings. Lastly, response on cloud worm was discussed and analysed. This research re-implemented OlexGA in cloud environment on the proposed EGA technique in order to check cloud worm detection's capability of genetic algorithm.