Enhancing Anomaly Based Intrusion Detection Techniques for Virtualization in Cloud Computing Using Machine Learning

By

Hina Batool

A Thesis Submitted to the Faculty of Department of Information Security, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment of the requirements for the degree of Master of Science in Information Security

August 2019

Declaration

I hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification either at this institution or elsewhere.

Acknowledgment

“Read In the name of your Lord (Allah), who created, created man from a clot. Read! And your Lord (Allah) is most bountiful. (He who taught) the use of pen taught man which he knew not.” (Al Quran 96:1-3)

I am thankful to my Creator Allah Subhana-Watala to have guided me throughout my thesis at every step and for giving me strength and ability to understand, learn and finally complete my thesis work.

I would like to thank my supervisor Mian Muhammad Waseem Iqbal for his expertise, ideas, feedback, time and constant encouragement.

Although no words could possibly quantify what a child owes to their parents, I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout in every department of my life. I would like to thank my parents for their utmost support, love and encouragement not only during my MS thesis but also during my entire life.

A special thanks to all my family, friends, NUST and Military College of Signals.

*Dedicated to my **mother** and **father** whose tremendous support and cooperation led me to this wonderful accomplishment.*

Abstract

Cloud computing is a rapidly growing technology that enables elastic, easy to manage, cost effective and on demand access to powerful computational resources for instance applications storage, servers, networks and services on internet. Convenient and ubiquitous deployment of clouds make it appealing to the users. The core component of cloud computing is virtualization, that is used to create multiple virtual machines from single physical machine in less cost. These Vms have a significant importance because of their ineludible utilization. Despite of all vital advantages, cloud computing domain is inclined to numerous security concerns. Intrusion is one such obstacle that can affect integrity and confidentiality of data. Host based intrusion detection systems have validated to be important in identifying network traffic and performance metrics of the guest machine maintained by hypervisor which do not conform to authorized patterns. In the current research, the aim is to make use of anomaly-based IDS to observe and recognize activities that are not part of legitimate network traffic in cloud computing environment using virtual machines. The proposed methodology observes the traffic of Virtual Machines. Scan and detect the anomalous behavior and in response restrain network traffic with anomalous packets.

Table of Contents

List of Tables

Chapter 1

Introduction

A rapidly growing cloud computing technology enables elastic, easy to manage, cost effective and on demand access to powerful computational resources for instance applications storage, servers, networks and services on internet. Convenient and ubiquitous deployment of clouds make it appealing to the users. There are different cloud services supplied to the end users or clients are basically of three distinct kinds which are SaaS namely Software as a Service, PaaS namely Platform as a service and IaaS namely Infrastructure as a service. Now these cloud services are deployed in four different models which are Community cloud, Private cloud, Hybrid cloud and Public cloud. Such classifications of different deployment schemes and services make it easier for users to attain the specifically suited services based on their organizational/personal demand.[1]

The core component of cloud computing is virtualization, In the field of computing, virtualization is creating a simulated adaptation of a machine or resource like a server, network or an operating system (OS) where the framework splits the subject into one or more execution environments [2]. Virtual machines (VMs) host individual OS, applications and services. This virtualized environment is created and managed by a hardware or software called Hypervisor [3]. Various VMs can be introduced on a solitary hardware as these are not dependent on the state of physical resource [1]. However, the segregation of VMs on a single machine has many security vulnerabilities and threats like virtual machine Sprawl, virtual machine Hyper jacking, VM Escape, virtual machine Theft, Incorrect virtual machine Isolation ,virtual machine Mobility, virtual machine to VM Attacks, VM Hopping, Flooding Attacks, Intrusion to Hypervisor, security of network, DoS attacks, Eavesdropping of Network Channel [4]. Intrusion detection mechanism is a useful methodology to mitigate threats associated with VMs in cloud computing [8].

Intrusion Detection System (IDS) is an efficient implementation for protecting networks from attacks or incursions. The fast developments in the expanding sector of virtualization has given an opening to prowlers, hackers and intruders the opportunity to discover various illegal

methods to exploit a network or machine [10]. As a result, the threat vector is also increasing with the evolution of fresh abilities and the main issue faced today is the strengthening of large volumes of network traffic data collected in network communication called ‘Big Data’ [9]. The Network Intrusion Detection System (NIDS) examines the intrusions by spotting the network traffic packets, affecting several network related hosts. NIDS has proven efficient against attacks on networks and a preferred option for networks to conquer many other techniques. Hence, a security analyzer must place this sensor at an appropriate area as it will affect its adequacy [5]. Anomaly based IDS (ABIDS) is an important variation in NIDS which identifies both new and novel attacks. ABIDS system utilizes an alternate philosophy which veers off from the signature based approach and identify the zero day, insider or new attack approaches. Nevertheless, as a reaction, it creates more noteworthy false alerts which really are not assaults. Therefore, a thoroughly trained system is crucial for achieving the required results [11].

Machine Learning is valuable in ABIDS with the end goal of recognizing and classifying attacks resulting in higher TP namely true positive and FP namely false positive rates. The methodology of parallel preparation is equally useful in the management of ‘Big Data’. Machine Learning centers around constructing computer programs and preparing them to evolve and alter when submitted to fresh data that can either be supervised: applying modifications to data that was made in the past or unsupervised: where deductions can be taken from datasets [10].

In this research, I have a proposed a hybrid intrusion detection technique that is the combination of both data mining and machine learning for prevention of malicious network traffic in virtualized cloud environment. Machine Learning technique is used to find patterns in data and then prediction of the outcome is done by using data mining technique.

1.1 Motivation and Problem Statement

In the current era of technology and development, security is one of the very basic and crucial factor to consider. None of the organization or even an individual person can compromise on the security. Computer security has its demand like never before. As the malwares and the intrusions has developed from being normal to next generation, accordingly, detecting and preventing them has evolved from normal to next generation with increase in complexity and computing power. Any unauthorized access can destroy the expected network flow and

jeopardize the basic objective of any network, hence intrusion detection has very crucial part when it comes to security.

Intrusion detection can be of signature or anomaly based, the anomaly based intrusion detection system is very effective approach in sensing attack patterns in a network traffic flow. There are certain set of problems related to the availability of Big data on network traffic on cloud which makes the intrusion detection a very difficult task in the first place. Intrusion detection technique either uses data mining or machine learning techniques which require some features or attributes as input for the processing/training purpose. The extraction of such feature vectors plays a vital role in detection accuracy but shortlisting the important features is nevertheless quite a tedious task.

Thus, the aim of this research is to provide an efficient intrusion detection system using machine learning technique so as to acquire required set of data and use the related patterns for detecting intrusions in real time basis.

1.2 Objectives

The main objectives of thesis are: -

- Analysis of different techniques for intrusion detection.
- Creating a cloud environment using virtualized environment for getting data set.
- Proposition of enhanced anomaly based intrusion detection technique for host IDS in virtualized environment of Cloud Computing.
- Implementation of detection technique which is the combination of both machine learning and data mining approach.

1.3 Relevance to National and Military needs

a. **National Needs**. The use of cloud computing in private, commercial and government sectors is on the peak. A great number of organizations are moving towards visualized environments because of its cost effectiveness and efficiency. No industry can progress

without such system that alerts them about the intrusions in their network. An efficient anomaly based intrusion detection system that can detect malicious attacks will play a vital role in prevention of these attacks.

b. **Military Needs**. Military data centers are provisioning services based on Cloud Computing. Having such systems, which are mostly working on virtualized environment, if the securities of these systems are compromised it will directly affect the security of our nation. A detection system that can effectively detect malicious attack can improve the security of such systems

1.4 Thesis Organization

The rest of the study is arranged as follows:

- In chapter 2, Background and literature review is discussed.
- Chapter 3 gives the complete methodology adopted for setting up experimental test bed
- Proposed Model and all necessary components of my research are described in chapter 4.
- Chapter 5 covers Results and comparative analysis.
- While Chapter 6 concludes the document.

Chapter 2

Background and Literature Review

An overview of Cloud Computing, Virtualization, Intrusion Detection System and Machine Learning along with the literature review will be presented in this chapter.

2.1 Cloud Computing

As defined by NIST, Cloud computing is a model for provisioning services in a very convenient way and provide access to the network on demand to a set of configured computing resources that includes resources like Servers, Networks, Services, Applications, and Storage devices. The edge that cloud computing has is based on its convenient and rapid deployment which can be done with the minimal of resources and minimal interaction with the service provider[1].

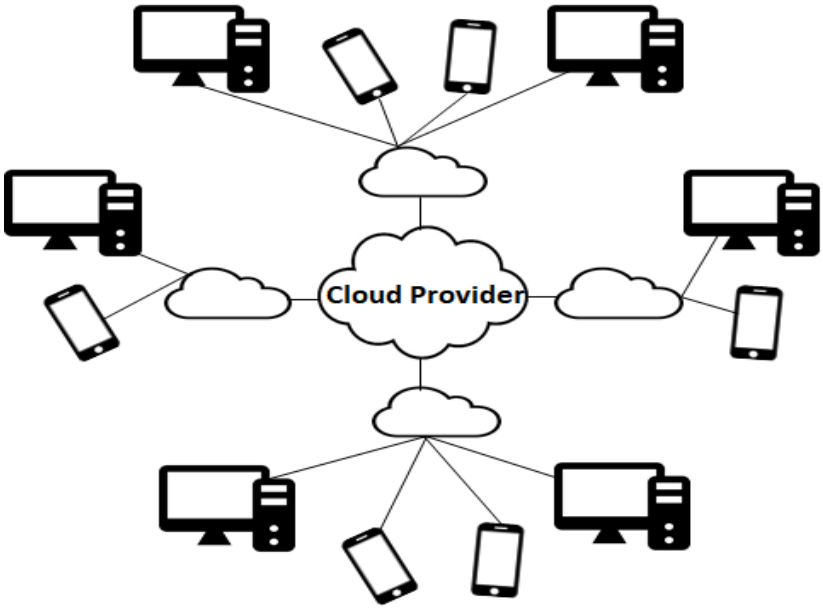

Figure 1: Different users connected with cloud system

2.1.1 Deployment Models of Cloud Computing

Models of cloud computing are basically categorized into four models: Public, Hybrid, Private, and Community cloud.[16]

a. Public Cloud Environment

A Public cloud providers make their resources available to public for free of cost, anyone can access and use the resources for their own purpose without any restriction. A cloud with such particularity runs on the provider premises which owns and manage while operating at the same time, their services of cloud. The location remains separated from their customer and has none physical control over the infrastructure. Such service providers, offer free and paid models along with the storage and computing. It is inexpensive in compare to the other cloud models.[16]

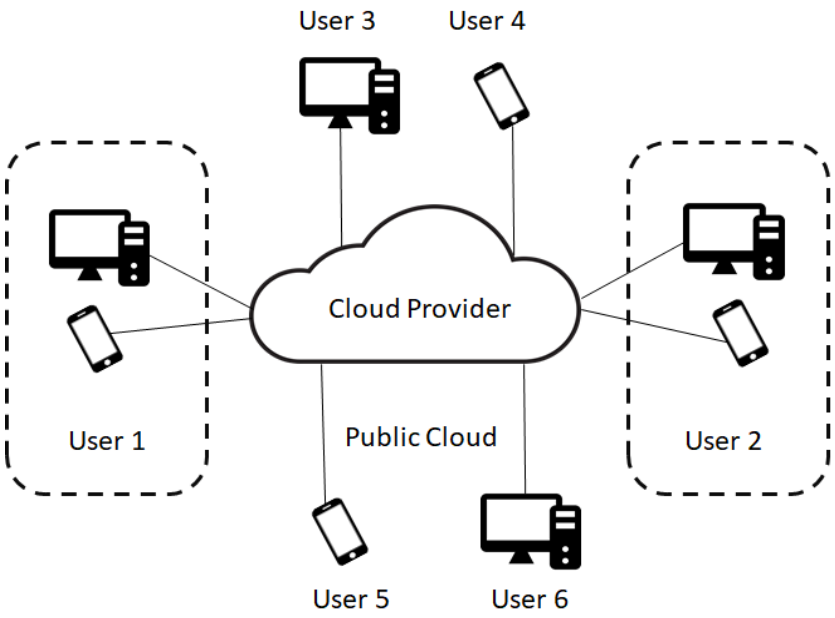

Figure 2: Public Cloud Environment

b. Hybrid Cloud Environment

Hybrid cloud provides the flexibility to the users to access both the private cloud and the public cloud resources from a single point of management environment. Hybrid cloud environment combine services and data from many different cloud models to substantiate an automated and a properly managed computing environment. Secondly, cloud providers also provide an atmosphere that is workable and effective in perspective of all possible situations. There may be a circumstance where customers use a private cloud for regular workloads but access one or more internal clouds as and when needed for high-demand work. The system's safety space and facilities are governed by role and private policies[2]. Figure 2.5 shows an environment in the Hybrid Cloud.

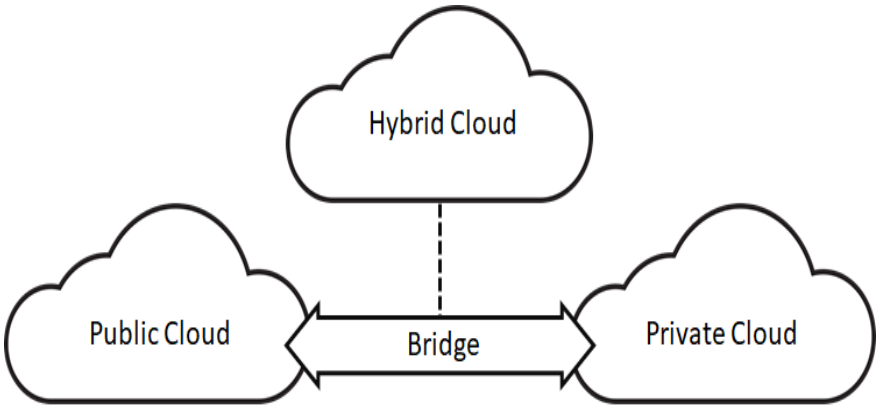

Figure 3: Hybrid Cloud Environment

c. Private Cloud Environment

A private cloud has emerged to fulfill the requirements of users of a single organization. Such infrastructures are owned and managed by the respective operated organizations within their own premises. Private cloud provides scalability and self-service. Private clouds do not have to be in the same location.[16]

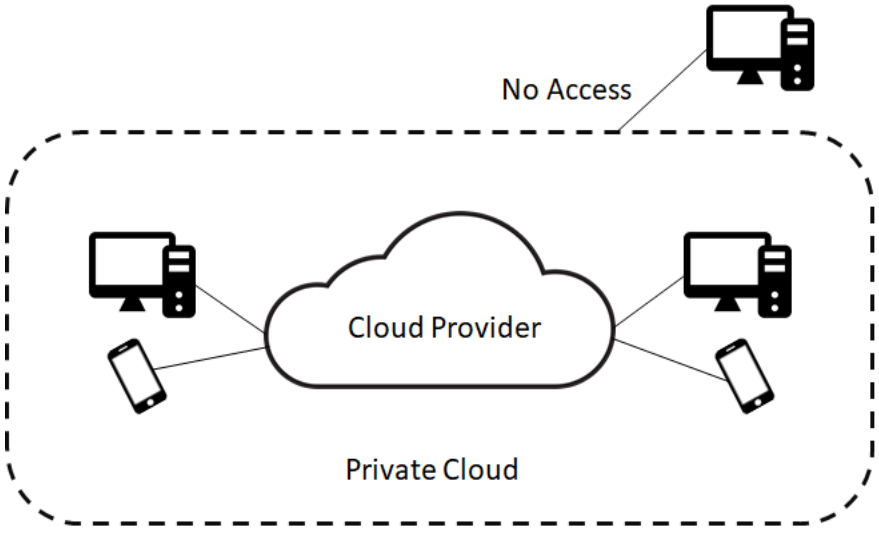

Figure 4: Private Cloud Environment

d. Community Cloud Environment

This cloud environment is exclusive use by a group of users. The users can share security requirements, common mission, policy or specific regulation or compliance they need to follow. It can be located Off premise or on premise. On-site community cloud computing is the method to implement the cloud environment by some community members. This situation is worth sharing alternatives as customers from various participating organization accessing common computing asset group. For a community cloud to be operational, at least one community member must introduce the cloud services. In terms of safety, each participant organization will have its own safety perimeter and linked via a secure communication link [1]. Figure demonstrates community cloud environment.

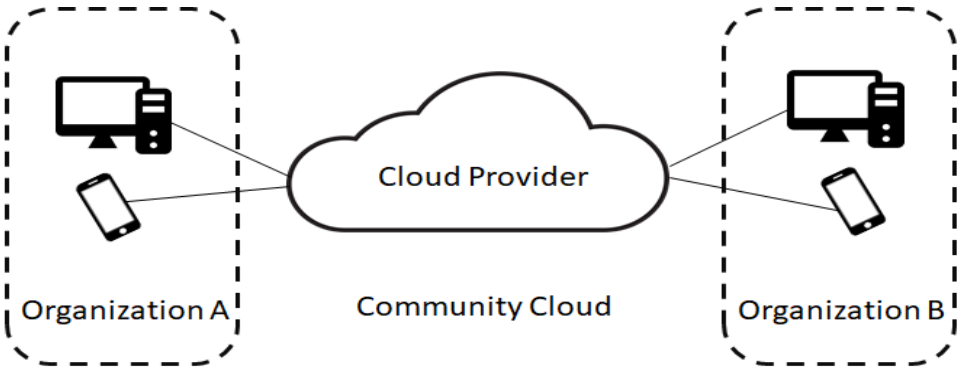

Figure 5: Community Cloud Environment

2.1.2 Services of Cloud Computing

The cloud computing model consists of three services, each of which is a specific resource in a cloud computing environment. These services explain the delivery of cloud services to customers. This model comprises of three services comprising of Software, Platform, and Infrastructure, as a service. Figure shows a cloud computing services pyramid.

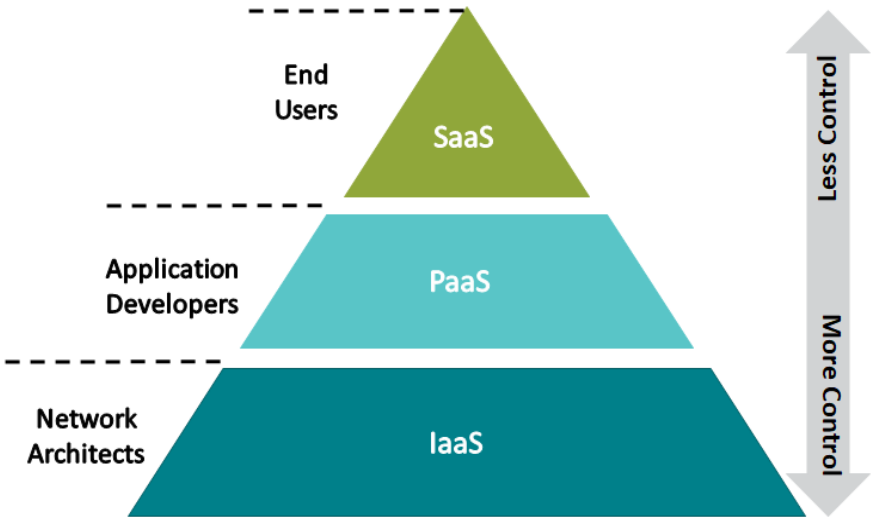

Figure 6 : Cloud Computing Model Stack

a. Software as a Service

Software as a service (SaaS) is a kind of service that ask users to pay as per the number of user's demand to use the application. Unlike to the license bought program that we see most of the time, this service provides an independent platform and user do not need to install software. Such cloud computing is inexpensive due to the fact that SaaS runs only an instance of software and readily makes it available for many of the end users. Vendor manages the entire computing resources which are responsible for delivering SaaS. Its globally accessibility from any kind of platform makes it excellent for collaborative working.[6]

b. Platform as a Service

PaaS namely Platform as a service is primarily a development environment and it consist of web server, an operating system, database and programming language execution environment. All of these features engulf the environment where the users run their program without the underlying knowledge of infrastructure after building, compiling and running their code. In such models, users basically manage application resources and the data.[16]

.

c. Infrastructure as a Service

Infrastructure as a Service (IaaS) make reference to services provision by the cloud service provider by distributing hardware infrastructure into smaller virtual machines including processor, the network, memory and numerous other resources. Instead of buying infrastructures such as network equipment, servers, related software, data center space and hardware, customers subscribe on demand to a fully outsourced service. The entire infrastructure is given to the client by the cloud provider to operate his application including all dedicated software and hardware in a manner that allows the client to manage different load circumstances. In IaaS, it is the sole responsibility of the cloud provider to manage all infrastructure[2]

2.2 Virtualization

Operation framework in typical PC frameworks offers level of observation over the equipment, on which many procedures can run simultaneously. This empowers a solitary operating system to work on single framework (hardware) on a given time.

A Hypervisor is a piece of software running next to or under an operating system intended to be an effective and isolated duplicate of the actual (physical) machine. Furthermore, a single hypervisor can also run on multiple networks hardware systems. However, Hypervisors related to single hardware system will be discussed.

Hypervisors or Virtual Machine Monitors(VMM) are of two types, bare metal and hosted. Type 1 VMM is installed directly on hardware and has full control over all VMs using it. Type 2 VMM sits beneath an OS above the hardware.

2.2.1 Motivating Factors for Security using Virtualization

Confidentiality, integrity and availability being the prime components of computer security are achieved through their properties of a virtualized system which are termed as isolation, oversight and duplication respectively.

Isolation, covering confidentiality is achieved by placing OS inside a virtual machine thus separating software running on the OS and hardware itself. This technique allows few risky services to be run in a virtual machine[9]whereas critical ones on other.

Oversight and duplication which covers integrity and accessibility are accomplished by utilizing a property of VMM called introspection which can catch and reestablish framework state and in this way an exceptionally helpful property of VMs.

2.2.2 Security in Virtualization

Security covers disclosure an alteration of data which involves, but not limited to the availability, integrity and confidentiality of:

- Software Data
- Software and Hardware Operational State
- Control and Network channels

Secure layout of any system needs accomplishing a threat model comprising of following processes, vision, diagram, validate, identify threats and mitigate threats.

2.2.3 Security Threats arising from properties of Virtualization

There are certain security implications are needed to be deliberated before configuring a fully virtualized system. Security properties of a virtualized system can have double effect enhancing as well as compromising security. These threats [9] occurred due to Hypervisor’s trust model, transparent nature and introspection capabilities.

The major vulnerability area in a hypervisor system is the blind trust on its platform. Usual VMs cannot detect the hypervisor being fully transparent and this inability can have drastic consequences for some softwares.

VM insertions, introspection, VM cloning, non-linear VM operations and software delinking from physical and hardware environment are few of the security threats that can create following issues in a virtualized environment.

- Inability to locate a VM
- Hidden or coward VM
- Additional consolidated hardware
- Physical location issues

2.2.4 Security Consequences from weakly Implemented System

Many security consequences hail from improper, incomplete and compromised implementation of virtualization requirement. Two primary kinds are transparency breaches and resource control breaches. Virtualization transparency is breached [9] if any of the three conditions is breached. This leads to information disclosure which deduces the presence of VMM. As VMM is the most important part of virtualized systems, compromised VMM puts on risk the whole system. If VMM software is interrupted, this will cause interruption to all Virtual machines running upon it. Any alteration to VMM can affect all VMs and also poses the risk to underline hardware.

2.2.5 Secure Implementation of Hypervisor System

System administrator must focus on rollout planning and management problems, hardening threat prevention and detection measures for vulnerabilities [9]recovery and measures for intrusion prevention and detection and measures for continuous protection. These items are compatible with any secure system implementation however hardening and threat prevention has particular concern for system virtualization platform.

2.3 Intrusion Detection

Any unwanted or illegal pursuit in a computer network can be a network intrusion. This is also called an attack on the network. In general, these type of attacks can be active or passive.

Passive attacks influence a confidentiality of a network but do not infringe the state of the system. A computer network is scanned, controlled or monitored for the ports that are open and

vulnerabilities in passive attacks. These type of attacks have significantly small likelihood of detection. Both passive and active scanning are classified in passive attacks. In this instance, an intruder does not actually interact with the system to obtain the required target system data, while in active recognition an attacker usually performs a port scan in active recognition. Passive attacks can include social engineering, eavesdropping, analyzing traffic, port scanning etc.[16]

Active attacks infringe a network's confidentiality, integrity and accessibility, thus changing the targeted system. In the event of active attacks, an intruder or hackers attempt to change target information or information that is enrouting to the targeted system.

An Intrusion Detection System (IDS) [8] gathers and examines data from a computing system on continuous basis, the main aim is to detect any intrusions. IDS has two main types working with analyzed data: Network-based Intrusion Detection System (NIDS) and Host-based Intrusion Detection System (HIDS)

2.3.1 Network-Based Intrusion Detection System

A NIDS examines and verifies network traffic on each layer of the stack for Open Systems Interconnection. It also decides and analyzes the planned purpose of traffic for untrustworthy activities. An alert is produced when an attack is identified or an unusual conduct is detected. Unlike HIDSs, most NIDSs are installed on a network and can scan traffic simultaneously from many devices.

The NIDS is implemented at a strategic stage or various points in the network where firewall is situated to conduct its prescribed operation function from all devices on the network. It scans the network-wide passing traffic against the threats of any attacks. If an attack is recognized or any unusual conduct is detected an alert is produced. Although NIDS scans all outbound and inbound traffic that could sometimes generate a bottleneck affecting the speed of network.[15]

NIDS can be categorized as off-line and on-line. Offline NIDS treats stored data by passing it through certain processes to determine whether or not it is an attack, while Online NIDS treats the network in real time by analyzing the packets to determine any malicious behavior.

2.3.2 Host-Based Intrusion Detection System

A Host-based Intrusion Detection System (HIDS) is an IDS category that screens a single computer's security from external and internal attacks. It inspects for any suspicious activity that occurs in a single host.

Internal attacks may be case in which detection is particular to inspect which resource has been accessed by which program and whether are there any security interruptions while in external attacks HIDS examines packets coming to and from that system. Mainly, HIDS most likely works in the same way as many virus scanners. HIDS response mechanism is based on activity logging and alert generation. These systems are intended to check and assess network traffic along with various system configurations that includes local log audits, local security policy, software calls and much more. HIDS needs to be deployed on each device and must be configured to a specific operating system and software.[15]

2.3.3 Intrusion Detection System Detection Methods

Methods used to evaluate gathered data to detect intrusions can be categorized in: signature detection and anomaly detection.

2.3.3.1 Signature Based IDS

Signature-based IDS relies on the database of attack signatures that are already known and generates an alarm if any malicious network activity is observed after matching the signatures in the database. This detection methodology has an elevated detection rate against known attacks but is not efficient against new attack detection. It is therefore essential to update the signature database to detect latest attacks.

Many IDS use the signature-based methodology and operate in the same way as a virus scanning tool. It looks for a recognized signature for each detection event. While signature-based IDSs are quite effective in sniffing renowned attacks, the entire method relies on periodic signature updates to its optimum results, much like an antivirus software. In other words, signature-based IDS efficiency is great based on its stored signature database.[18]

2.3.3.2 Anomaly Based IDS

Anomaly-based IDS uses features of system and network to model typical network behaviors. Any divergence from the typical patterns of traffic is regarded as an attack. Anomaly-based IDS reveals intrusive attacks and abnormal information systems operations. In a dynamically evolving environment, these kinds of systems work. The objective stays the same to recognize all real attacks properly while recognizing non-attacks adversely. It thus, continues an efficient strategy for identifying and reacting to malicious operations in networking and computing resources.

Unlike the signature-based strategy, anomaly-based IDS does not depend on predefined signatures to detect attacks and thus novel attacks can be identified. As a downside, they are described by an elevated false positive (FP) rate, in which even valid traffic can sometimes be reported as attacks. In addition, a comprehensive review is needed to determine whether traffic is a real attack and what it is achieving. Utmost care in implementing an anomaly-based IDS to detect attacks is therefore be practiced. Anomaly detection is used for our work.[11]

2.4 Machine Learning

Machine learning can be described as a mechanism in which computer systems are constructed in such a manner that a learning system is implemented and experience improves automatically. It is the computer science subfield that provides the skill of computers to learn without explicit programming. Pattern Recognition and Artificial Intelligence studies can trace the evolution of machine learning. It concentrates on developing computer programs which have the skill to train themselves to evolve and alter when subjected to new information. It does so by studying and building algorithms through learning and predicting the information.[21]

Machine learning is strongly linked to computational statistics and data mining while being strongly linked to mathematical optimization. It is a very powerful predictive analysis method that helps analysts generate accurate and repeatable choices and outcomes while revealing the hidden perspectives in the information.

2.4.1 Machine Learning Algorithms Classifications

Machine learning algorithms are classified as supervised, unsupervised or semi-supervised learning.

2.4.1.1 Supervised Learning

Supervised learning is the task of concluding a function from the labeled training data. in supervised learning, the output data sets are given so that they can train the machine to achieve the required output. There is input variable (x) mapped to a function by using a learning algorithm to produce the output variable (y) in a way that

$$Y=f(x)$$

The objective is to predict the mapping function well enough that by using the algorithm to predict the yield for the next input data. The algorithm uses a sequence of iterations to predict the raining data which is then rectified by the function. After an acceptable level of performance, the learning method stops.

Regression and classification are the two types that are grouped into supervised learning. If the output variable is a real value such as a person's money or weight, it is referred to as a regression problem whereas if the category of the output variable is alike color being blue or red or a state such as unhealthy or healthy then it is mentioned to as a classification problem.

2.4.1.2 Unsupervised Learning

Unsupervised learning is the job of concluding a function from unlabeled data to define hidden structure. The data is clustered in different classes in unsupervised learning without the need to use datasets. The data is accessible as input data (x) in this situation, however, there are no respective output variables available. The objective here is to model a basic structure in the data in order to gain understanding or to discover more about the data. There are no correct answers in this approach rather the algorithms are left to find and present structures in the data at discretion.

Association and clustering are the two types that are grouped into unsupervised learning. Association issues are associated with finding rules that explain data components whereas clustering issue is used when it is necessary to discover the intrinsic grouping of data.

2.4.1.3 Semi-Supervised Learning

Semi-supervised learning is the job of concluding a function to define hidden structure from a great section of unlabeled data and a slight amount of labeled data. This classification lies between the unsupervised and supervised learning. Unlabeled data can be utilized as training data by using supervised learning problems to predict new data while unsupervised learning methods can be used in input variables to assess and learn the structure.

In summarizing, the entire data is labeled in supervised learning and the algorithms learn to predict the output from the input and entire data is unlabeled in unsupervised learning and the algorithms learn the inherent structure from the input data, lastly most data is unlabeled while some is labeled in semi-supervised learning, hence, a combination of unsupervised and supervised methods are used.

2.5 Related Work

A.M.Gadal et al. proposed “Anomaly Detection Approach using Hybrid Algorithm of Data Mining Technique”. In this research, a hybrid machine learning technique has been proposed for network based intrusion detection based on union of Kmeans clustering and Sequential Minimal Optimization (SMO) classification, with high accuracy. Moreover, it also aims to maximize positive detection rate and minimize and false negative alarms in networks.[8]

M. S. Dildaret al. presented “Effective Way to Defend the Hypervisor Attacks in Cloud Computing “.The authors have proposed the Virtual Machines and Hypervisor Intrusion Detection System in detection and prevention of attacks of hypervisor in virtualized cloud computing domain. This technique has embraced numerous features from other methods by frequently checking the tasks which then avoids suspect occurrences.[9]

Xiaohua Li proposed a joint machine learning and human learning design approach to make the training data labeling tasks in linear regression problems more effective and vigorous to noise, modeling mismatch and human labeling errors in which they applied active learning for

the search of better quality training data and used outlier detection process for the removal of human labeling errors. They also derived thresholds in order to remove data which was prone to errors to keep the sparsity of the labeling data [12].

Ye, X., Chen, X., Wang, H. presented An Anomalous Behavior Detection Model in Cloud Computing The author has developed anomaly based detection model in virtualized cloud computing environment. This model captures inter-Virtual Machine traffic flow, detect unknown and known abnormal network behaviors, select hybrid methods to analyze control network systems and VM network behaviors. [10]

Asry et al. made an endeavour to detect anomalous behaviour in network traffic using Classification and Regression Tree (CART) and Fuzzy Logic in KDD Cup' 99 dataset [14]. They used CART to build rules or models and those models were implemented using Fuzzy inference engines. Fuzzy logic was used in performing tests and the resulting rules were used for classification to produce desired results. They proposed that CART combined with Fuzzy Logic can be effectively used to build a classifier that can be helpful for observing anomalies in an intrusion detection system. The average accuracy obtained through their experiments was considerable to gain from the hybrid effect of both utilized techniques.

J. Lin et al. proposed Automated Anomaly Detection and Root Cause Analysis in Virtualized Cloud Infrastructures. The author has presented a technique for automatically detecting anomaly and root cause inspection in virtualized cloud data centers. In the first stage, to identify abnormal system behaviors, they used unsupervised learning techniques. Then a mechanism for root cause inspection with consideration to anomaly generation between units of system has been proposed. Finally, they have used actual test bed of virtualized cloud, which shows their technique effectively recognizes abnormal system behaviors and determines their reasons accurately. [11]

In [15], Kayvan Atefi et al. used Visual Architecting Process methodology to evaluate different algorithms and hybrid models used for intrusion detection. They further used the essential parameters to compare the results. The idea proposed was to utilize different techniques to evolve the best process for finding accurate and acceptable results in IDS thus reducing false positives and false negative alarms in the network.

Spanaki Pet al. presented "Cloud Computing: Security Issues and Establishing Virtual Cloud Environment via Vagrant to Secure Cloud Hosts". This research proposes a distributed

intrusion detection system for Cloud computing environments. The suggested Intrusion Detection System consists of five main modules. The suggested Intrusion Detection System is implemented on the Google Cloud Platform and CIDDS-001 public dataset is used for testing. Using an algorithm of time-based sliding window, captured network traffic is processed on each cloud router and passes it to an anomaly detection module identify attack of each kind. [2]

Hossein et al. proposed an Intrusion Detection model using Genetic Algorithm (GA) in [16]. They combined developmental and traditional algorithms to accelerate search speed thereby reducing the false alarm rate on training dataset. They constructed classifying models based on the training data observed from the GA taking K-Nearest Neighbors (KNN) Algorithms into consideration.

Scheme	Dataset	Attacks	Mechanisms	Result
Network Intrusion Detection System Using various data mining techniques[22]	NSL-KDD	DOS, Probe, R2L, U2R	This research paper includes the implementation of different data mining algorithms including Linear regression and K-Means Clustering to automatically generate the rules for classify network activities	Accuracy with Linear Regression 80% , K-Means Clustering 67.5%
Anomaly Detection Approach using Hybrid Algorithm of Data Mining Technique.[8]	NSL-KDD	DOS, Probe, R2L, U2R	This paper proposed a hybrid machine learning technique for network intrusion detection based on combination of Kmeans clustering and Sequential Minimal Optimization (SMO) classification. Comparison between the proposed approach (K-mean + SMO) and individual algorithm K-mean clustering and Sequential Minimal Optimization (SMO) classification has been done	Accuracy rate is 97.3695%, Positive detection rate is 94.48% and reduced the false alarm rate is (1.2%)
Machine Learning	AWID	Flooding	In this paper authors have	Accuracy with OneR

Techniques for Intrusion Detection on Public Dataset[23]		, Probe, Injection	evaluated Aegean Wi-Fi Intrusion Dataset (AWID) with different machine learning techniques. OneR, Ada Boost, J48 Decision tree, Random Forest and Random Tree machine learning techniques are used to evaluate the AWID dataset with information gain and chi squared statistics based feature selection.	92.07%, with J48 92.21%, with Random Forest 92.29%, with Random Tree 90.76% and with Ada Boost 91.80%
Implementation of classification and regression Tree (CART) and fuzzy logic algorithm for intrusion detection system[14]	KDD Cup 99	DOS, Probe, R2L, U2R	The author has made an endeavour to detect anomalous behaviour in network traffic using Classification and Regression Tree (CART) and Fuzzy Logic. They used CART to build rules or models and those models were implemented using Fuzzy inference engines. Fuzzy logic was used in performing tests and the resulting rules were used for classification to produce desired results	Based on various test cases, they gained the highest accuracy rate of 85.68%
Anomaly Detection Based on Profile Signature in Network Using Machine Learning Technique[15]	KDD Cup 99	DOS, Probe, R2L, U2R	The researchers used Visual Architecting Process methodology to evaluate different algorithms and hybrid models used for intrusion detection. The author uses GA and SVM	The highest AC for the whole features using the hybrid model of SVM and GA was 98.333%.

Table 1: Schemes presented in different researches

The efficacy of DARPA and KDD datasets have sublimed over a period of time. This aspect has been addressed in our research by incorporating present day attacks and protocols in the developed dataset. The precise application of machine learning algorithm on this dataset to achieve desired results is distinctive feature of our work.

Chapter 3

Setting Up Experimental Test Bed

3.1 BIOS Setup Utility

Following steps are used to access BIOS Setup Utility:

- Power-on or power-cycle the server.
- F2 key is pressed, while the system is performing POST.
- Attach a bootable USB containing operating system and set boot priority.

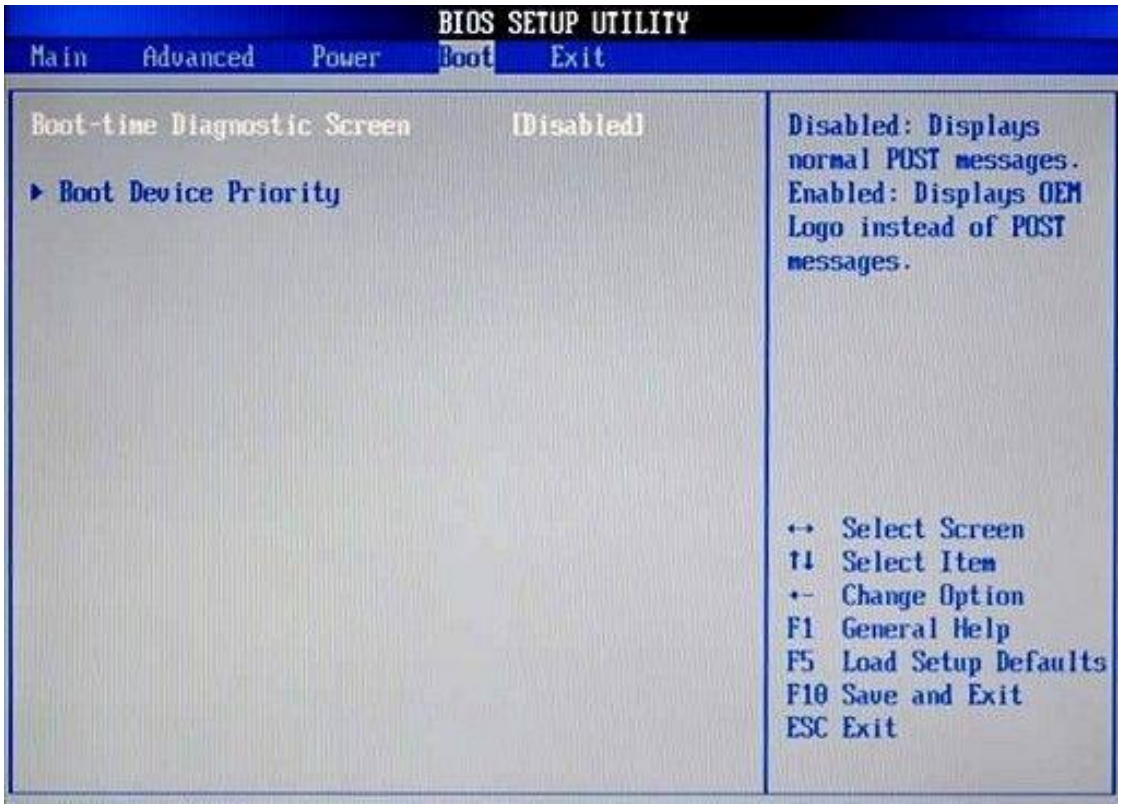

Figure 7: BIOS Setup Utility Console

3.2 Installing ESXi

Installations of ESXi was interactive as well as scripted, and many of the options were there to boot installer and after the installation then access the media installed.

3.3 Storing the ESXi Installation Script and Media

USB flash or external hard drive can be used to store ESXi installation script media. That drive can be used later for the installation of script for ESXi. When there are several USB drives attached to the installation machine, that particular machine which has the installation software, starts to look for installed script and media on all the attached USB ports in which USB drives are attached. The important point to be noted here is the use of flash drive for media installation which has been done separately and so as boot device installation.

3.4 Pre-requisites

To obtain USB with ESXi installation script and media, we require the following enlisted items:

- ESXi image (ISO format)
- Installation script (Kickstart file)
- Portable USB drive

3.5 Installing ESXi Using the Interactive Mode

ESXi CD/DVD was used for the installation of ESXi software on the hard drive of server.

- Insert the ESXi 6.7 Installable media through bootable USB
- Restart is required after the bootable USB is attached to machine
- Need to go to BIOS settings to change the preference of installation from any default to USB device
- As the machine restarts, press the Enter key on the initial screen where the installation continues
- Go through the user license agreement and to accept the agreement we need to press F11 key

Figure 8: System Boots After the Restart

Figure 9: EXSi Installation Begins

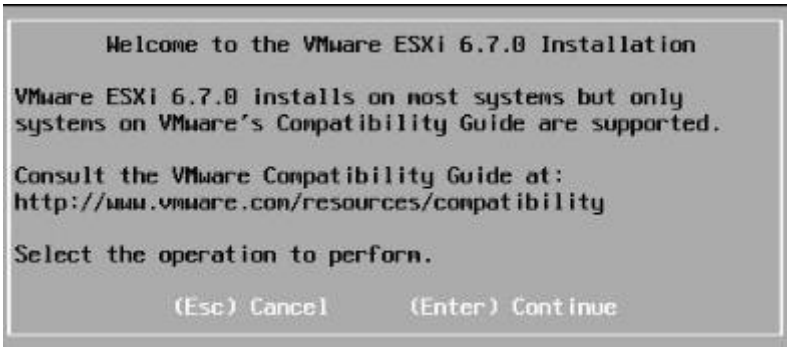

Figure 10: For the continuity of installation press enter

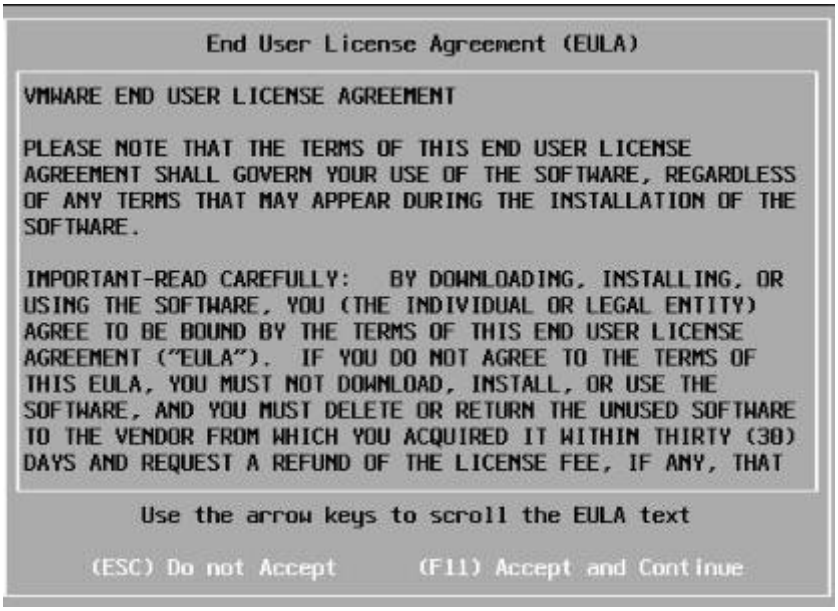

Figure 11: Accepting User License is Mandatory for the Installation to Complete

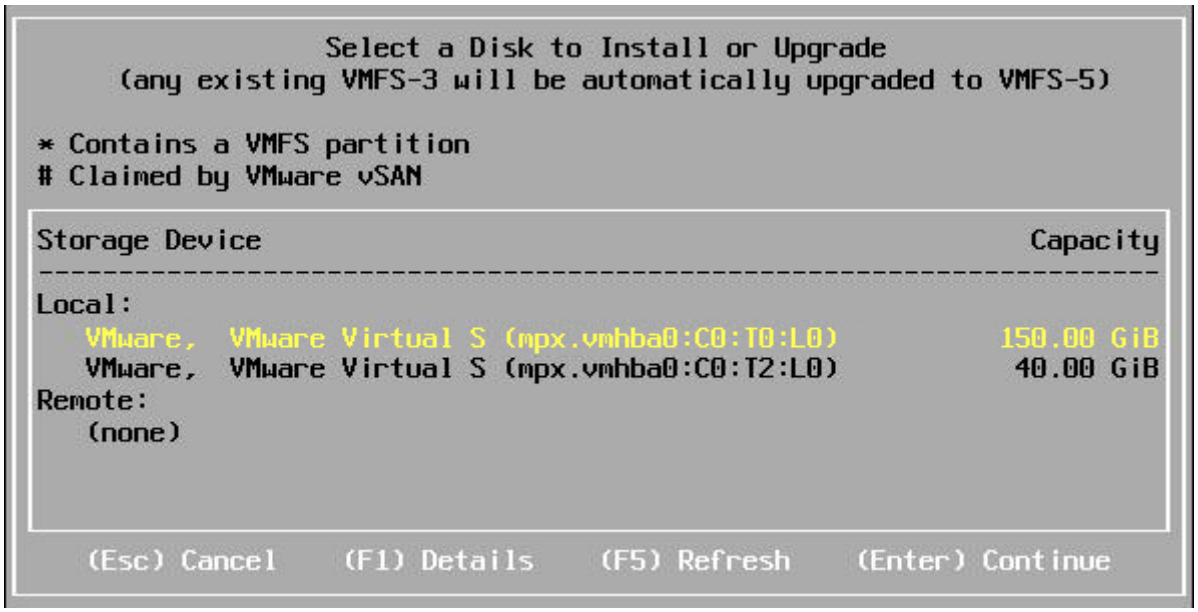

Figure 12: Selecting Hard disk/Media for Installation

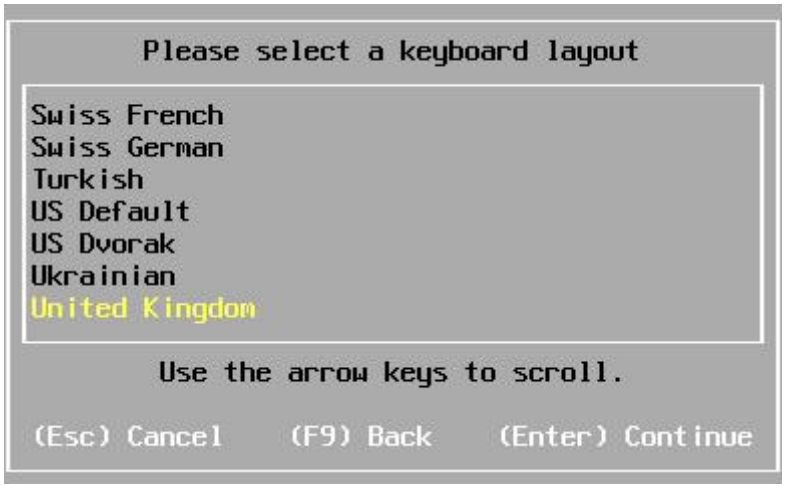

Figure 13: Selecting of User Language

Figure 14: Setting of Root Password for Administrator Account

Figure 15: System will Reboot After the Installation is Complete

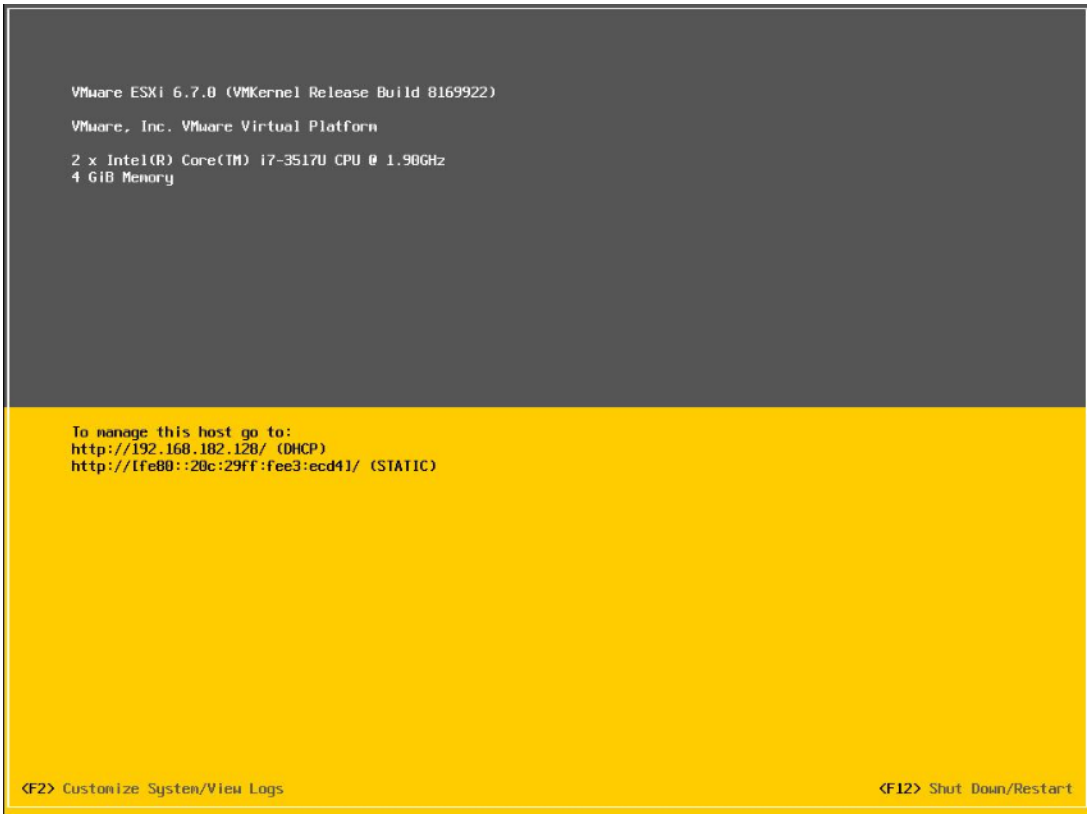

Figure 16: This Screen shows a successful Completion of Installation

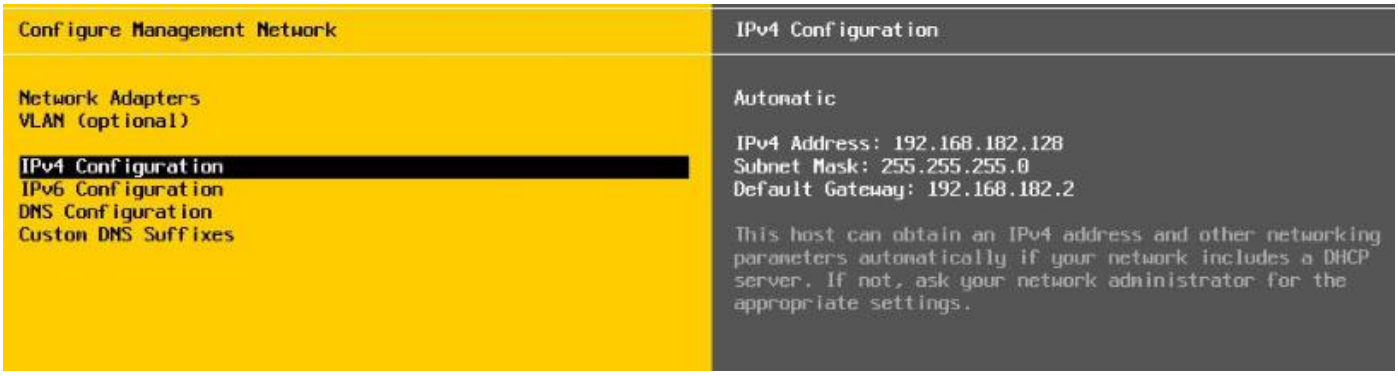

Figure 17: Go to Configure Management Network for changing Network Parameters

Chapter 4

Proposed Enhanced Technique

4.1 Intrusion Dataset Generation and Collection

4.1.1 Tools Used

a. Metasploit Framework using MSFconsole

"The msfconsole is probably the most popular interface to the Metasploit Framework (MSF). It provides an "all-in-one" centralized console and allows you efficient access to virtually all of the options available in the MSF."

b. Kali Linux

Kali Linux may be a Debian-based Linux distribution geared toward advanced Penetration Testing and Security Auditing. Kali has many hundreds of tools for varied information security tasks, like Penetration Testing, Reverse Engineering and Forensics. Kali Linux is founded, developed and maintained by Offensive Security, a number one info security coaching company.

4.1.2 Intrusion Dataset

The experimental dataset was developed utilizing an ESXi server. Attacking virtual machine and Victim Virtual Machine was established in a simulated environment.

Figure 19 illustrates the data set generation process. The step wise process is explained as under:

4.1.2.1 Data Logging in Hypervisor

ESXi 6.7 was used to simulate Victim virtual machine comprise of Windows 10 operating system and attacking virtual machine comprise of Kali Linux operating system. The network logs

were collected over a period of four days for three weeks. Required data was collected in PCAP format using the builtin pktcap-uw packet capturing tool in ESXi 6.7. Pktcap-uw tool compared to the tcpdump-uw tool is enhanced version for packet capture and analysis. The legacy tcpdump-uw tool can only sniff or capture packets or frames at the vmkernel level. Tcpdump-uw tool cannot sniff packets or frames at the vSwitch level or uplinks level, or virtual ports level. The latest pktcap-uw tool has greater flexibility when it comes to the packet capturing since it can capture traffic at every level within hypervisor which results in enhanced level of troubleshooting. The metasploit framework and Kali Linux were used to generate attack traffic. Latest attack information was obtained from Common Vulnerabilities and Exposures(CVE) website. Our data set has 6 main attacks categories including Reconnaissance, DoS, Fuzzers, DDoS, Shellcode and Backdoors along with the normal network traffic. The Hypervisor was then setup to obtain the logs of all network activity for the prescribed time period using pktcap-uw by enabling ESXi SSH console.

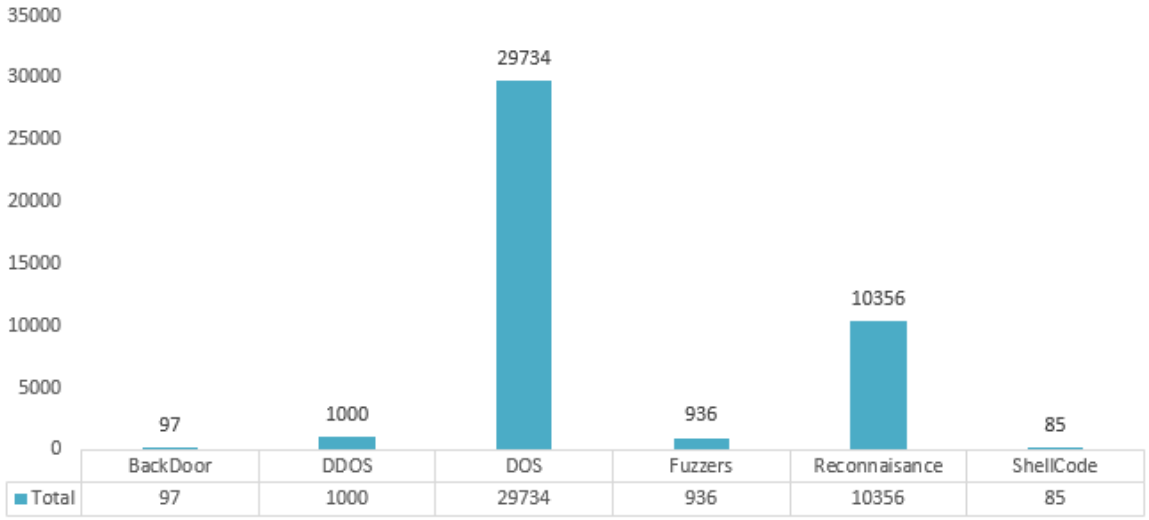

Figure 18: Total records of different attacks

Type of Attack	Total Count
Reconnaissance	10356
Fuzzers	936
DoS	29734
BackDoor	97
DDoS	1000
Shellcode	85
Total	**42208**

Table 2: Total Count of Different Attacks

4.1.2.2 Dataset Feature Extraction

Logs of the Hypervisor were used to create PCAP files using the ESXi pktcap-uw packet capturing tool. We have extracted 50 features from the dataset which are Start Time, Last Time,Count, Duration, Average Duration, Source Address Destination Address, Prototype, Source Port, Destinatiton Port, Bytes, IpId, Stoss, Dtoss, Sttl, Dttl, Sbytes, Dbytes,Spkts, Dpkts, sload, dload, sloss, dloss , Src_pps, Dst_pps, SourceId, Ind, Mac Address, Destination Address, Dir, Sjitter,Djitter, Status, User, Swin, Dwin, Packets, Trans, Sequence, Svlan, Dvlan, Smpls, dmpls, Length, Info,Checksum,WindowSize, AckNum, ProtocolNum and Attack Type using wireshark and Argus which are network traffic auditing and monitoring tools along with the self-written code in C#. These attributes are in the form of basic, flow, content, connection, time and labeled attributes. Furthermore, the outcome of the dataset was exported to CSV format for further processing.

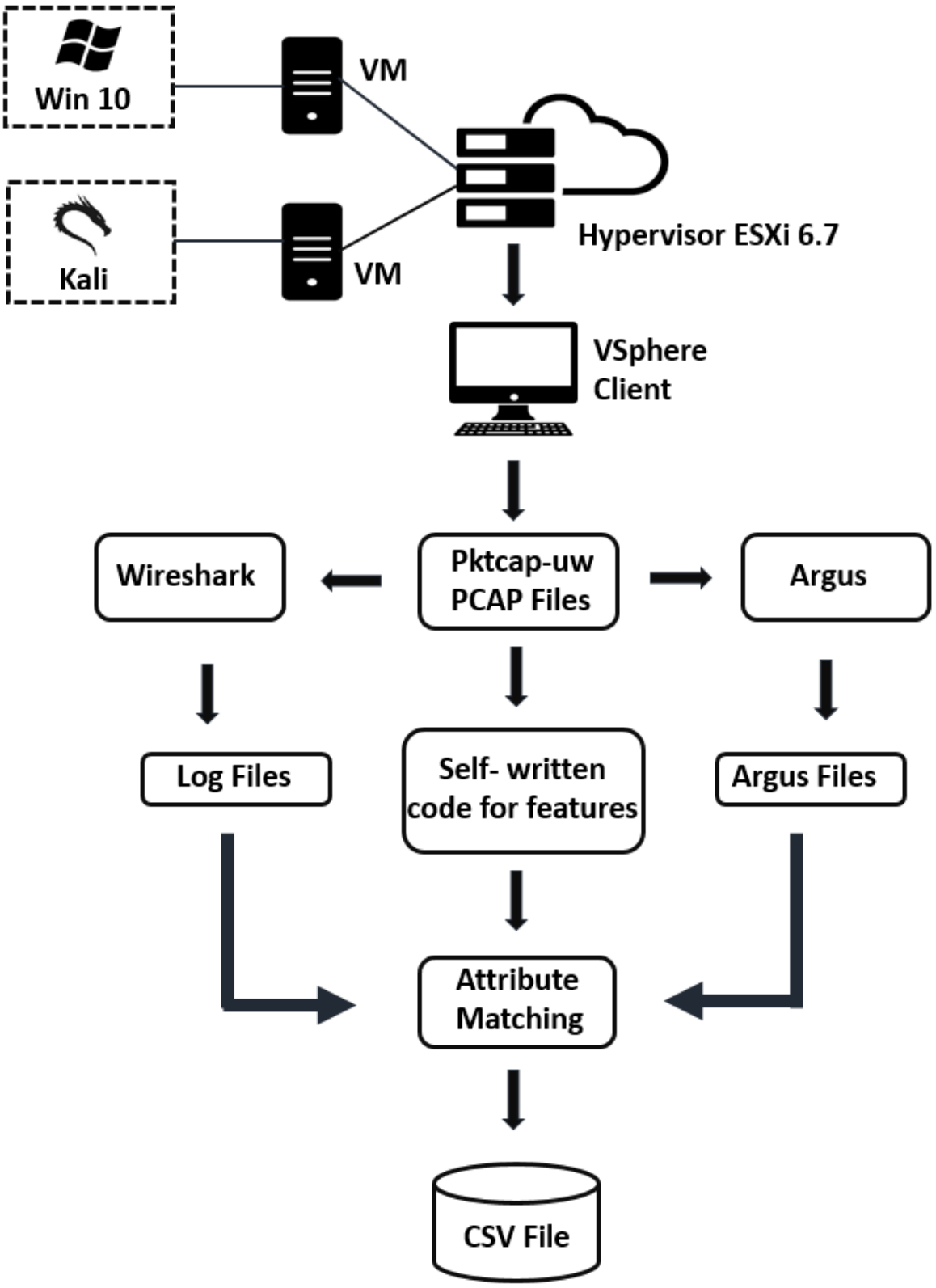

Figure 19: Dataset Generation

4.2 Proposed Model

Proposed model is mainly divided into three stages being Data Preparation,Data Pre-processing and Detection of Intrusion Detection by using classifiers. The proposed model is depicted in Figure20. The stage wise implementation is described as under:

4.2.1 Data Preparation

The logs contained in the prepared dataset was utilized for our proposed model. These logs were analyzed and a total of 50 attributes for each record were listed. Table 3 shows the Attack and Normal record distribution of dataset.

Serial	Detail	Total Count
1.	Normal Records	62826
2.	Attack Records	42208
	Total	**105034**

Table 3: Total Count of Normal and Intrusion Records

4.2.1.1 Normal / Attack Feature Vectors

The logs of the dataset was created for both normal and attack traffic. The vectors from both traffic were collected separately.

4.2.1.2 Labelling

In this stage, the normal and attack vectors were flagged by labelling as normal or attack type record so that the user could easily understand it.

4.2.1.3 Mixing

After labelling both records were combined in a single pool and displayed as a table from the csv file.

4.2.2 Data Pre-processing

The data pre-processing stage is combination of two processes, Feature Selection and Data Filtration.

4.2.2.1 Feature Selection

Feature selection is a procedure to address the problem by selecting a subset that is useful to a problem. Important features are selected to extract from dataset. Features that are unrelated to the problem are removed. The features that are important and improves the accuracy of the model are selected. Total 13 features are selected which are Dur, SrcAddr, DstAddr, Prototype, Sport, Dport, IpId, DstTOS, DstTTL, SrcJitter, DstJitter,Seq, AttackType.

4.2.2.2 Filtration

In Data Filtration step the selected features are scrutinized for any incomplete or missing records. Then these records are filtered out and removed from the data to avoid processing flaws of the next module.

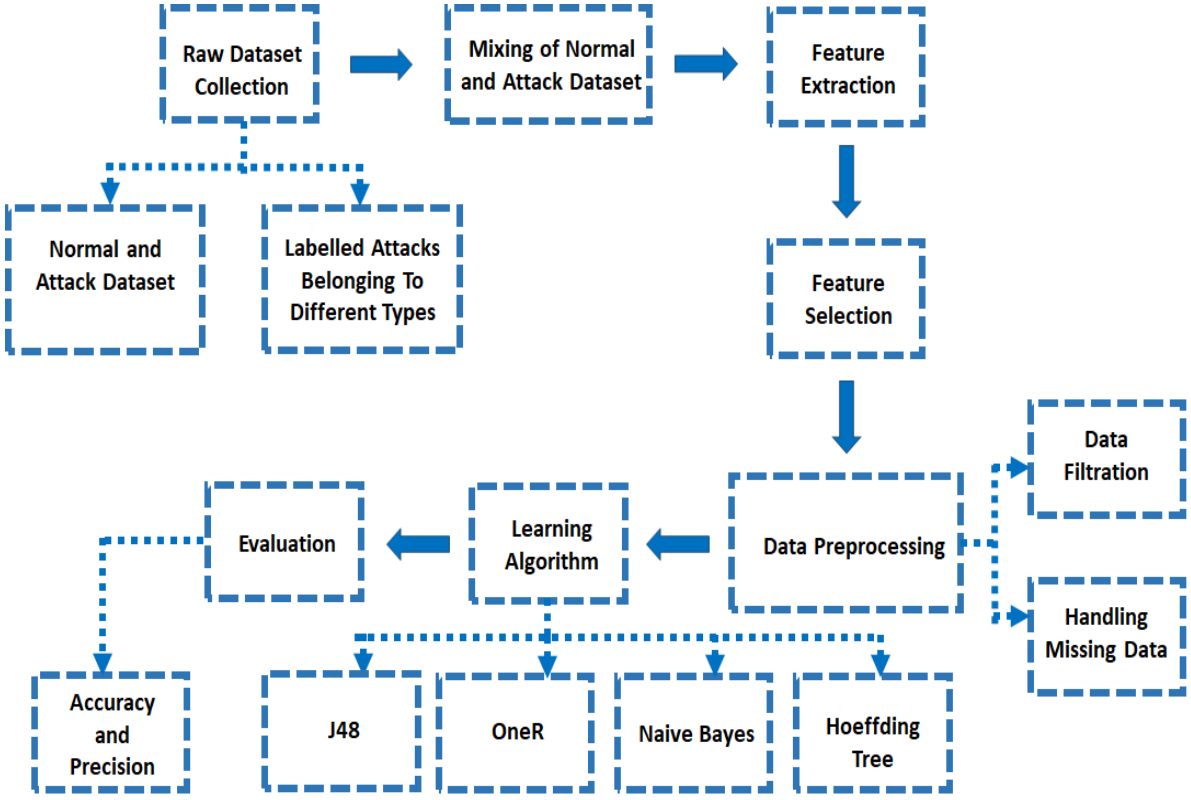

Figure 20: Proposed Intrusion Detection Architecture

4.2.3 Detection of Intrusion by using classifiers

Using classifiers to detect intrusion in virtualized cloud environment can be accomplished using a direct process. This process can take as little as few minutes or can be elongated to months, depending on the clarity of the objectives and scope, availability of dataset, and the pre-processing trials related with the data. Two rudiments of the analysis are collection of data and tool acquisition. The collected data entails a pre-processing stage to move it into the form which is required for classifier implantation and Intrusion detection. Result execution and analysis of data is a significant step to comprehend the subsequent model and its rule sets.

4.2.3.1 Result Execution and Analysis of Data

For the result execution and analysis an open source weka tool has been used. Weka is a best known data mining tool and provides a wide-ranging list of machine learning algorithms. The created dataset of 105022 records comprising of 13 features are converted in to .arff format.

ARFF files known as Attribute-Relation File Format used to work with weka machine learning software.

a. J48 Classifier

By implementing J48 classification algorithm we get an accuracy rate of 99.496%.

```
Time taken to build model: 2.33 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      104483               99.4962 %
Incorrectly Classified Instances       529                0.5038 %
Kappa statistic                          0.9909
Mean absolute error                      0.002
Root mean squared error                  0.0322
Relative absolute error                  1.2987 %
Root relative squared error             11.4531 %
Total Number of Instances           105012

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.711    0.001    0.552      0.711    0.622      0.626    0.997     0.456     BackDoor
                 0.999    0.004    0.710      0.999    0.830      0.841    0.997     0.691     DDOS
                 0.986    0.000    0.999      0.986    0.993      0.990    1.000     0.999     DOS
                 0.989    0.000    1.000      0.989    0.995      0.995    1.000     0.996     Fuzzers
                 1.000    0.001    0.999      1.000    1.000      0.999    1.000     1.000     Normal
                 0.998    0.000    1.000      0.998    0.999      0.999    0.999     0.999     Reconnaisance
                 0.120    0.000    0.429      0.120    0.188      0.227    0.990     0.333     ShellCode
Weighted Avg.    0.995    0.001    0.996      0.995    0.995      0.994    1.000     0.996

=== Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
    69     0     0     0    16     0    12 |     a = BackDoor
     0   999     1     0     0     0     0 |     b = DDOS
     0   408 29326     0     0     0     0 |     c = DOS
     2     0     4   926     4     0     0 |     d = Fuzzers
     0     0     0     0 62814     0     0 |     e = Normal
     0     0    10     0     6 10340     0 |     f = Reconnaisance
    54     0     0     0    12     0     9 |     g = ShellCode
```

Figure21: J48 Classifier Result

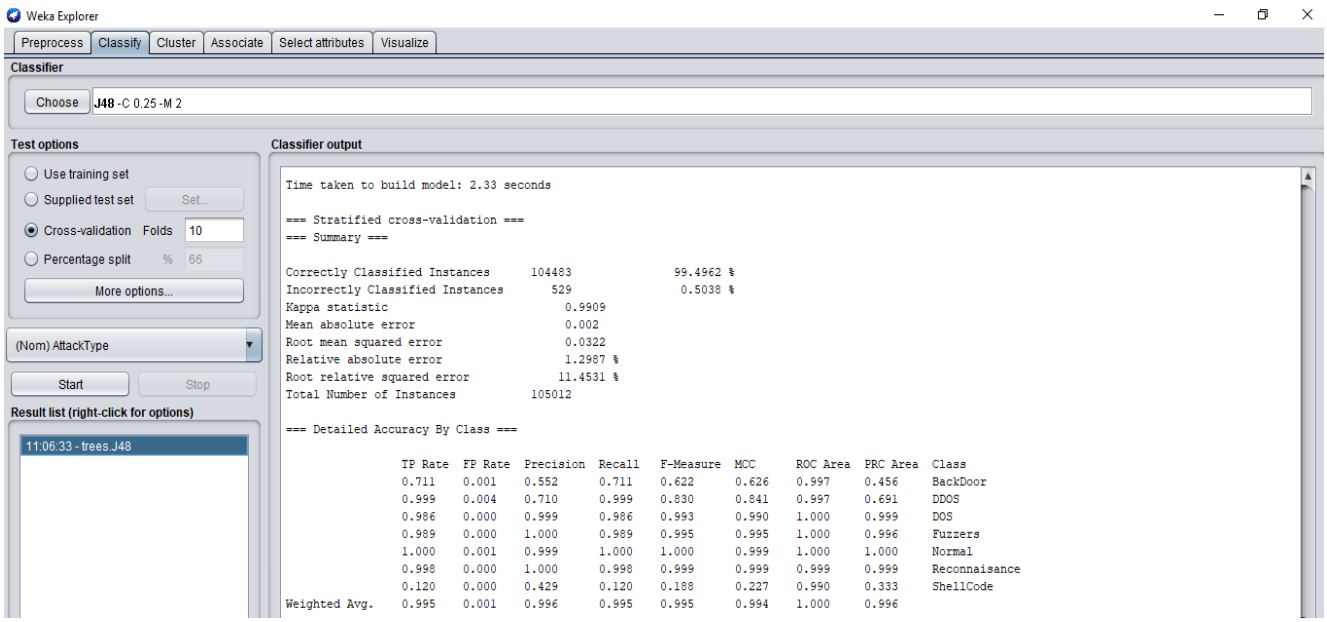

Figure 22: Result Continue

b. OneR Classifier

By implementing OneR classification algorithm we get an accuracy rate of 98.292%.

```
Time taken to build model: 0.29 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      103240               98.292  %
Incorrectly Classified Instances      1794                1.708  %
Kappa statistic                          0.9688
Mean absolute error                      0.0049
Root mean squared error                  0.0699
Relative absolute error                  3.0923 %
Root relative squared error             24.8694 %
Total Number of Instances           105034

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.093    0.000    0.409      0.093    0.151      0.194    0.546     0.039     BackDoor
                 0.785    0.000    0.950      0.785    0.860      0.863    0.892     0.748     DDOS
                 0.985    0.009    0.977      0.985    0.981      0.973    0.988     0.966     DOS
                 0.333    0.001    0.759      0.333    0.463      0.500    0.666     0.259     Fuzzers
                 1.000    0.022    0.986      1.000    0.993      0.982    0.989     0.986     Normal
                 0.969    0.000    0.999      0.969    0.983      0.982    0.984     0.970     Reconnaisance
                 0.021    0.000    0.222      0.021    0.038      0.067    0.510     0.005     ShellCode
Weighted Avg.    0.983    0.016    0.981      0.983    0.981      0.972    0.984     0.968

=== Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
     9     0     0     0    81     0     7 |     a = BackDoor
     0   785   204     0    11     0     0 |     b = DDOS
     0    30 29288    92   318     6     0 |     c = DOS
     0     9   467   312   139     9     0 |     d = Fuzzers
     0     0     0     0 62814     0     0 |     e = Normal
     0     2    27     7   290 10030     0 |     f = Reconnaisance
    13     0     0     0    82     0     2 |     g = ShellCode
```

Figure 23: OneR Classifier Result

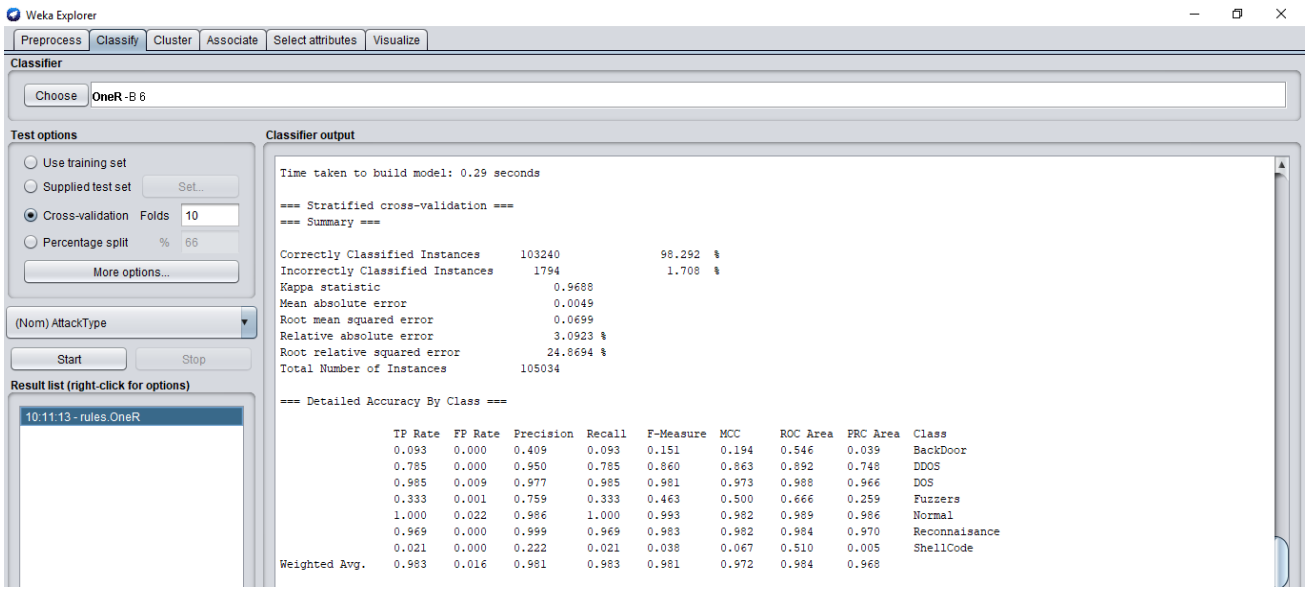

Figure 24: Result Continue

c. Naive Bayes

By implementing Naive Bayes classification algorithm we get an accuracy rate of 96.731%.

```
Time taken to build model: 0.15 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      101601               96.7315 %
Incorrectly Classified Instances      3433                3.2685 %
Kappa statistic                          0.9412
Mean absolute error                      0.0098
Root mean squared error                  0.0861
Relative absolute error                  6.221  %
Root relative squared error             30.6525 %
Total Number of Instances           105034

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.309    0.001    0.297      0.309    0.303      0.302    0.999     0.333     BackDoor
                 0.999    0.001    0.921      0.999    0.958      0.959    1.000     1.000     DDOS
                 0.989    0.033    0.922      0.989    0.954      0.937    0.999     0.998     DOS
                 0.081    0.000    0.962      0.081    0.150      0.278    0.998     0.818     Fuzzers
                 0.971    0.002    0.998      0.971    0.985      0.963    0.999     0.999     Normal
                 0.970    0.006    0.944      0.970    0.957      0.952    0.997     0.991     Reconnaisance
                 0.381    0.001    0.303      0.381    0.338      0.339    0.999     0.327     ShellCode
Weighted Avg.    0.967    0.011    0.969      0.967    0.964      0.947    0.999     0.995

=== Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
    30     0     0     0     0     0    67 |     a = BackDoor
     0   999     1     0     0     0     0 |     b = DDOS
     0    82 29417     0     2   233     0 |     c = DOS
     4     4   795    76    52     4     1 |     d = Fuzzers
     7     0  1437     0 60994   359    17 |     e = Normal
     0     0   266     3    39 10048     0 |     f = Reconnaisance
    60     0     0     0     0     0    37 |     g = ShellCode
```

Figure 25: Naive Bayes Classification Algorithm Result

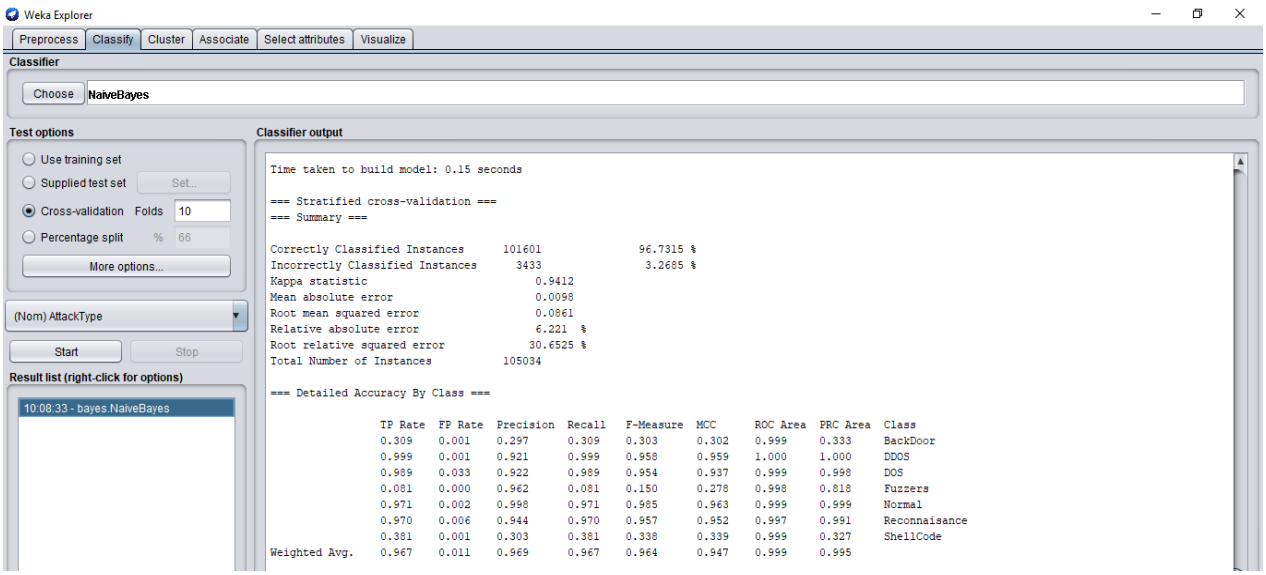

Figure 26: Naive Bayes Classification Algorithm Result Continue

d. Hoeffding Tree

By implementing Hoeffding Tree classification algorithm we get an accuracy rate of 94.697%.

```
Time taken to build model: 3.46 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       99464               94.697  %
Incorrectly Classified Instances      5570                5.303  %
Kappa statistic                          0.9022
Mean absolute error                      0.0715
Root mean squared error                  0.1755
Relative absolute error                 45.3285 %
Root relative squared error             62.4688 %
Total Number of Instances           105034

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.289    0.001    0.167      0.289    0.211      0.218    0.894     0.260     BackDoor
               0.000    0.001    0.000      0.000    0.000      -0.003   0.674     0.041     DDOS
               0.929    0.028    0.929      0.929    0.929      0.901    0.975     0.930     DOS
               0.946    0.001    0.936      0.946    0.940      0.940    0.984     0.956     Fuzzers
               1.000    0.047    0.969      1.000    0.984      0.961    0.997     0.998     Normal
               0.783    0.012    0.881      0.783    0.829      0.813    0.970     0.847     Reconnaisance
               0.196    0.000    0.373      0.196    0.257      0.270    0.854     0.327     ShellCode
Weighted Avg.  0.947    0.037    0.938      0.947    0.942      0.919    0.985     0.953

 === Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
    28     0     6    19    12     0    32 |     a = BackDoor
    81     0    60     0   840    19     0 |     b = DDOS
     3   108 27611    11   933  1068     0 |     c = DOS
     0     0    35   885     3    13     0 |     d = Fuzzers
     0     0     0     0 62814     0     0 |     e = Normal
     2    14  2007    22   204  8107     0 |     f = Reconnaisance
    54     0     3     9    12     0    19 |     g = ShellCode
```

Figure 27: Hoeffding Tree Classification Algorithm Result

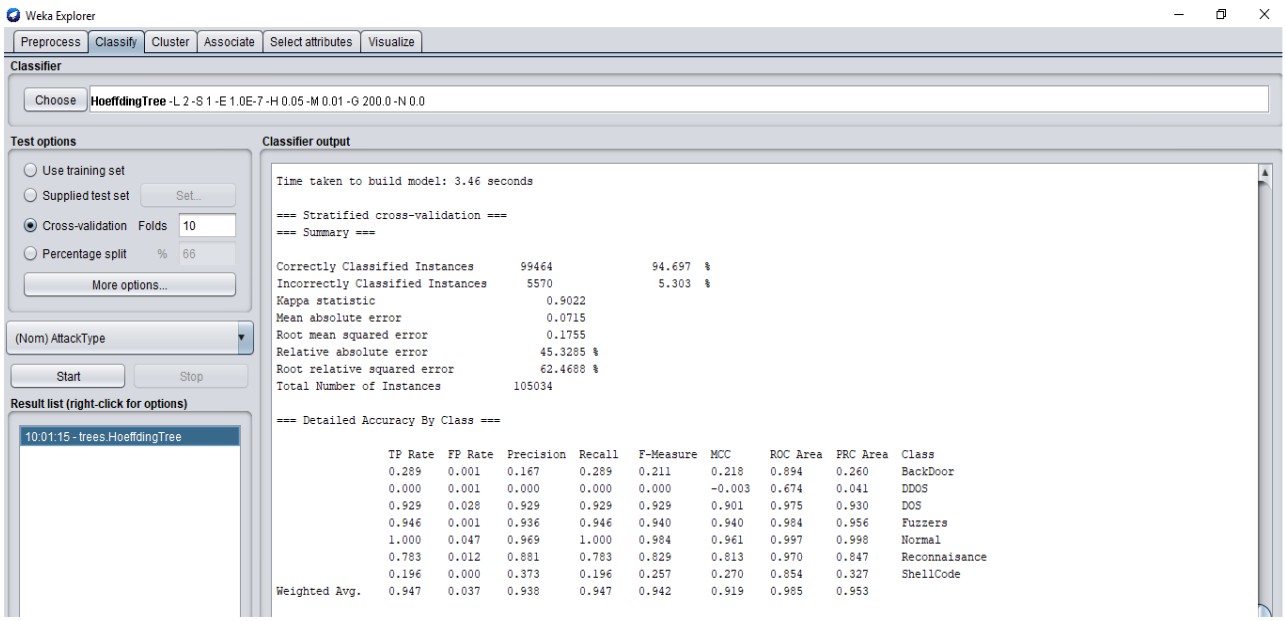

Figure 28: Hoeffding Tree Classification Algorithm Result Continue

4.2.3.2 Conclusion

Implemented data mining classification algorithms are used to classify new portions of information into predefined groups. Classification algorithms use pre-classified dataset to classify data, based on current trends and patterns. After rule generation, the logic from the implemented algorithm can be combined into numerous intrusion detection technologies including firewalls and IDS signatures. Out of 4 implemented classifiers J48 algorithm shows highest accuracy rate of 99.496%

Classification Algorithm	**J48**	**OneR**	**NaiveBayes**	**Hoeffding Tree**
Accuracy	99.496%	98.292%	96.731%	94.697%

Table 4: Table Accuracy Rate Comparison of Classifiers

Chapter 5

Result and Comparative Analysis

5.1 Introduction

To validate the performance of our proposed system 10-Fold cross validation is implemented. Cross validation computes the accuracy of the implemented model by dividing the dataset into training testing set. J48,OneR, Naïve Bayes and Hoeffding Tree classification models are created from the training set and their accuracy is calculated grounded on how well it classifies the testing set.

5.2 10-Fold Cross Validation

Three classifiers OneR, Naïve Bayes and Hoeffding Tree are trained and tested with 10 fold cross validation i.e., the created dataset is divided arbitrarily into 10 subsets, where 1 subset is used for testing and 9 for training. For every combination the process is repeated 10 times. This procedure aids in assessing the strength of a given approach to detect malwares that exploits heap based overflow vulnerability without any previous information.

For evaluation of the propose system the following quantities are considered:

- **True Positives (TP):** Number of intrusions that exploit virtualized cloud environment are classified as intrusions.

- **False Positives (FP):** Number of benign programs classified as Intrusions that exploit virtualized cloud environment.

Classification Algorithm	**J48**	**OneR**	**NaiveBayes**	**Hoeffding Tree**
Accuracy	99.496%	98.292%	96.731%	94.697%
TP	0.995	0.983	0.967	0.947
FP	0.001	0.016	0.011	0.037

Table 5: Accuracy Rate of Classifiers with TP and FP

Implementation of 10-Fold Cross Validation technique has significantly increase the accuracy rate of the system. The comparison of the modeled data is given below

Classification Algorithm	J48	OneR	NaiveBayes	Hoeffding Tree
Accuracy (10-Fold Cross Validation)	99.496%	98.292%	96.731%	94.697%
Accuracy (Percentage Split)	98.358%	97.898%	95.285%	88.150%

Table 6: Comparison of Accuracy rate before and after Data Modeling

5.3 Comparative Analysis

We compared our results with [24], their accuracy rate with J48 Classifier is 89.672% with NSL-KDD dataset. In [25] researchers performance of algorithms is tested on KDD99 dataset. Their accuracy with J48 classifier is 96.9% .The accuracy rate of both the researches are less as compared to our accuracy with J48 classifier which is 99.496% with our dataset. It is noteworthy that NSL-KDD and KDD99 dataset contained only four major attacks whereas in our case there are six major attack vectors. [19] produced their own dataset that contains benign and seven attack network flows. Their highest accuracy rate with ID3 classifier is 98% and came out 88% accuracy with NaiveBayes which is less as compared to our accuracy rate achieved with this classifier which is 96%.Results of these researches are shown in Fig 29.

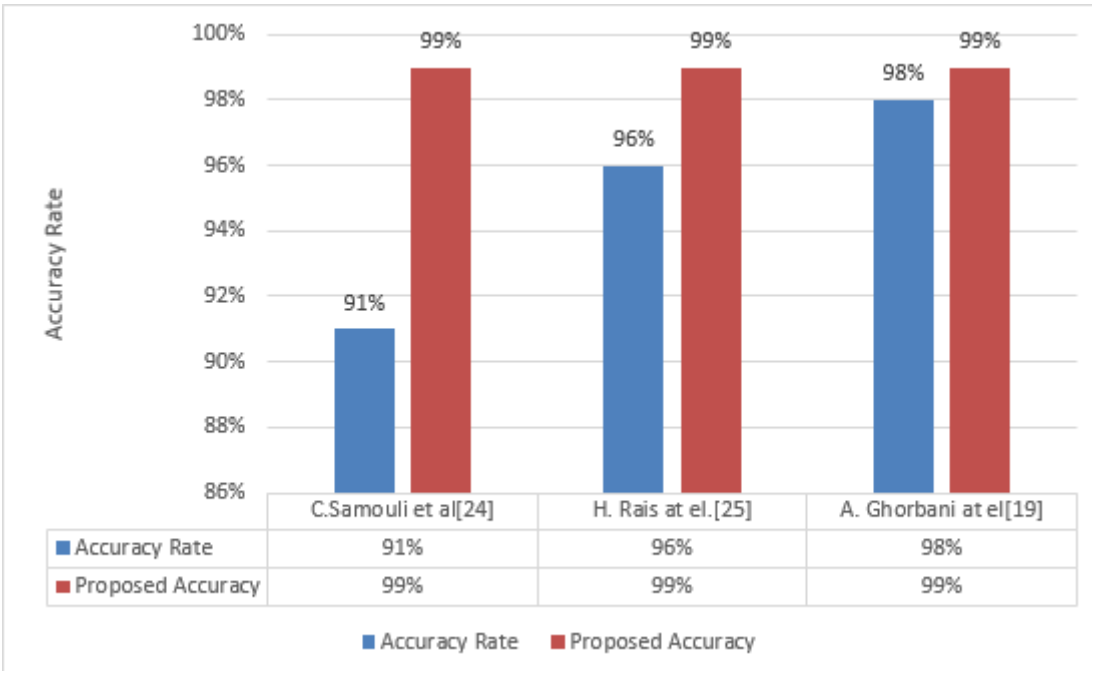

Figure 29: Result Comparison

5.4 Conclusion

By the implementation of proposed system a comprehensive detection technique is presented for classification of attacks that intrudes in virtualized cloud environments.

The logic from the implemented algorithm could be used to upsurge awareness of Advance Persistent Threats strategies and advance the complete security of the organizations. Details of our work is concluded in this chapter and future work is discussed in detail in next chapter.

Chapter 6

Conclusions and Future Work

With the invention of latest computing resources, the performance of intrusion detection is improving by multiple folds. This enhancement and continuous research helps in the advancement of the performance of intrusion detection systems. These advancements not just facilitate users in detecting the malicious content but also open a vast door of opportunities for intruders to exploit the weakness and improve their malicious content to a level where intrusion detection system fails to classify them as malicious.

In this research, we have presented Intrusion Detection system using combination of data mining and machine learning techniques for intrusion detection in virtualized cloud environment. The main achievement of this research was the creation of our own data set, which can be further used for development of new systems and research. The created feature set is used to train three classifiers OneR, Naïve Bayes and Hoeffding Tree for the detection of intrusions in virtualized cloud environment. The proposed methodology is easy to implement in operations of cyber security to comprehend the behavior of malwares targeting their organizations. In future this dataset is expected to be useful to the research community of network intrusion detection system in virtualized cloud environment and to be observed a contemporary network intrusion detection system standard dataset.

References

[1] M. Ali, S. U. Khan, and A. V. Vasilakos, “Security in cloud computing: Opportunities and challenges,” Inf. Sci., vol. 305, pp. 357– 383, 2015, doi:10.1016/j.ins.2015.01.025.

[2] Spanaki P., Sklavos N. Cloud Computing: Security Issues and Establishing Virtual Cloud Environment via Vagrant to Secure Cloud Hosts. In: Daimi K. (eds) Computer and Network Security Essentials. Springer, Cham, 2018, pp. 539-553. (In Eng.) DOI: 10.1007/978-3-319-58424-9_31

[3] Asraa Abdulrazak AliMardan and Kenji Kono. “Containers or Hypervisors, Which is Better for Database Consolidation?”. In IEEE 8th International Conference on Cloud Computing Technology and Science, pp 564-571, 2016.

[4] Andreas Blenk, Arsany Basta,Martin Reisslein andWolfgang Kellerer. “Survey on Network Virtualization Hypervisors for Software Defined Networking”. In IEEE Communications Surveys & Tutorials (Vol: 18, Issue: 1, First quarter), pp 655-685, 2016.

[5] S. A. Hussain, M. Fatima, A. Saeed, I. Raza, and R. K. Shahzad, “Multilevel classification of security concerns in cloud computing,” Appl. Comput. Informatics, vol. 13, no. 1, pp. 57– 65, 2017.

[6] Cheraghlou, M. N., Khadem-Zadeh, A., & Haghparast, M. (2016). A survey of fault tolerance architecture in cloud computing. Journal of Network and Computer Applications. 61(1). 81-92.

[7]S. Mahdi Shariati, Abouzarjomehri and M. Hossein Ahmadzadegan. “Challenges and Security Issues in the Cloud Computing from two perspectives: Data security and privacy protection”. In 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), 2015.

[8]Saad Mohamed Ali Mohamed Gadal, Rania A. Mokhtar, “Anomaly Detection Approach using Hybrid Algorithm of Data Mining Technique”, 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE).

[9] M. S. Dildar, N. Khan, J. B. Abdullah, and A. S. Khan, Effective way to defend the hypervisor attacks in cloud computing, in Proc. 2017 2nd Int. Conf. Anti-Cyber Crimes, Abha, Saudi Arabia, 2017, pp. 154–159.

[10] Ye, X., Chen, X., Wang, H., et al.: 'An anomalous behavior detection model in cloud computing', Tsinghua Sci. Technol., 2016, 21, pp. 322–332.

[11]J. Lin, Q. Zhang, H. Bannazadeh, and A. Leon-Garcia, "Automated anomaly detection and root cause analysis in virtualized cloud infrastructures," in Proc. IEEE/IFIP Netw. Oper. Manag. Symp. (NOMS), Istanbul, Turkey, 2016, pp. 550–556.

[12]Xiaohua Li and Jian Zheng. "JointMachine Learning andHuman LearningDesign with Sequential Active Learning and Outlier Detection for Linear Regression Problems". In Annual Conference on Information Science and Systems (CISS), pp 407-411, 2016.

[13]Wahab OA, Bentahar J, Otrok H, Mourad A (2018) Towards trustworthy multi-cloud services communities: a trust-based hedonic coalitional game. IEEE Trans Serv Comput 11(1):184–201.

[14]Asry Faidhul Ashaari Pinem and Erwin Budi Setiawan. "Implementation of Classification and Regression Tree (CART) and Fuzzy Logic Algorithm for IDS". In 3rd International Conference on Information and Communication Technology (ICoICT), pp 266-271, 2015.

[15]Kayvan Atefi, Saadiah Yahya, Amirali Rezaei and Siti Hazyanti Binti Mohd Hashim. "Anomaly Detection Based on Profile Signature in Network Using Machine Learning Technique". In IEEE Region 10 Symposium (TENSYMP),pp 71-76, 2016.

[16]M. Hossein Ahmadzadegan, Ali Asgar Khorshidvand and Mehdi Ghalbi Valian. "LowRate False AlarmIntrusionDetection Systemwith Genetic Algorithm Approach". In 2nd International Conference on Knowledge based Engineering and Innovation (KBEI), 1045-1048, 2015.

[17]Geetanjali Nenvani and Huma Gupta. "A Survey on Attack Detection on Cloud using Supervised Learning Techniques". In Symposium on Colossal Data Analysis and Networking (CDAN), 2016.

[18]Abusitta A, Bellaiche M, Dagenais M (2018) A trust-based game theoretical model for cooperative intrusion detection in multicloud environments. In: 2018 21st conference on innovation in clouds, internet and networks and workshops (ICIN). IEEE, pp 1–8.

[19]Wahab OA, Bentahar J, Otrok H, Mourad A (2018) Towards trustworthy multi-cloud services communities: a trust-based hedonic coalitional game. IEEE Trans Serv Comput 11(1):184–201.

[20]Minh Tuan Pham, Phuc Hao Do and Kanta Tachibana. "Feature Extraction for Classification Method using Princial Component based on Conformal Geometric Algebra". In International Joint Conference on Neural Networks (IJCNN), 2016.

[21]Kayvan Atefi, Saadiah Yahya, Amirali Rezaei and Siti Hazyanti Binti Mohd Hashim."Anomaly Detection Based on Profile Signature in Network Using Machine Learning Technique". In IEEE Region 10 Symposium (TENSYMP), 2016.

[22] Gupta, D., Singhal, S., Malik, S., & Singh, A. (2016, May). Network intrusion detection system using various data mining techniques. In Research Advances in Integrated Navigation Systems (RAINS), International Conference on (pp. 1-6). IEEE.

[23] Udaya Sampath K. Perera Miriya Thanthrige, Jagath Samarabandu, Xianbin Wang,‖ Machine Learning Techniques for Intrusion Detection on Public Dataset‖, IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2016.

[24] S. Choudhury and A. Bhowal, "Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection," in Proc. Int. Conf. Smart Technol. Manage. Comput., Commun., Controls, Energy Mater., May 2015, pp. 89–95.

[25] T. Mehmood, H.B.M. Rais, "Machine learning algorithms in context of intrusion detection" 3rd International Conference on Computer and Information Sciences (ICCOINS), IEEE (2016)