



VIRTUALIZATION PRIVACY CONCERNS IN CLOUD COMPUTING ENVIRONMENTS : A TAXONOMY



**MRS.SRIVIDHYA ELANGOVAN
MRS.JAYANTHI SAMPATH
MS.A.SONYA
DR.NALINI SUBRAMANIAN
DR.GOKULAKRISHNAN**

VIRTUALIZATION PRIVACY CONCERNS IN CLOUD COMPUTING ENVIRONMENTS: A TAXONOMY

Mrs. SRIVIDHYA ELANGO VAN

**Assistant Professor, Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology, Semmancheri, Chennai,
Tamil Nadu**

Mrs. JAYANTHI SAMPATH

**Assistant Professor, Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology, Semmancheri, Chennai,
Tamil Nadu**

Ms. A. SONYA

**Assistant Professor, Department of Information Technology,
B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai,
Tamil Nadu.**

Dr. NALINI SUBRAMANIAN

**Associate Professor, Department of Computer Science and Engineering,
Prathyusha Engineering College, Thiruvallur, Tamil Nadu.**

Dr. S. GOKULAKRISHNAN

**Assistant Professor, Department of Computer Science and Engineering,
Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya,
Kancheepuram, Tamil Nadu**



KALAIVANI PUBLICATIONS

ERODE.



First Edition: May 2021

Copyright with Authors

All Publishing rights (printed and e-book version) reserved with Kalaivani Publications. No part of this book should be reproduced in any form Electronic, Mechanical, Photocopy or any information storage and retrieval system without prior permission in writing, from **Kalaivani Publications**.

ISBN: **978-81-954343-0-5**

Price Rs. **580** /-

Published By:

16, Weekly market road,
Bhavani, Erode – 638 301.

Phone: 04256 – 6234667

Web: www.kalaivanipublications.com

Email: submit@kalaivanipublications.com

ACKNOWLEDGEMENT

We express our heartfelt prayers to almighty God.

We would like to extend our sincere thanks to our Family members, colleagues, friends and well-wishers who stood with us in all our struggles and successes to complete this work done.

This book wouldn't have been possible without the support and vision of editor's comment which allowed us to complete this work.

Our heartfelt thanks to publishers and their entire team who have taken, lots of pain to get this quality book. We would like to thank for the support and love of our family members for their encouragement and support.

Finally we express our deepest gratitude to all those who have helped us in completing this book.

- Authors

TABLE OF CONTENTS

CHAPTER: 1 CLOUD COMPUTING: A COMPREHENSIVE INTRODUCTION

1.1. INTRODUCTION	1
1.2. THE EVOLUTION OF CLOUD COMPUTING	1
1.2.1. Time-Sharing on Mainframe Computers	1
1.2.2 Peer-to-Peer and Client-Server Computing	3
1.3. THE PEER-TO-PEER MODEL	4
1.3.1. The Client-Server Model	4
1.4. GRID COMPUTING	5
1.5. UTILITY COMPUTING	6
1.6. VIRTUALIZATION	7
1.7. SERVICE-ARRANGED ARCHITECTURE	8
1.8. THE CLOUD COMPUTING PARADIGM	9
1.9. DISTRIBUTED COMPUTING MODELS	11
1.9.1 Private Cloud	11
1.9.2. Public Cloud	12
1.10. DISTRIBUTED COMPUTING SERVICES	13
1.10.1 Infrastructure as a Service [IaaS]	14
1.11. PLATFORMS AS A SERVICE [PaaS]	16
1.11.1 Software as a Service [SaaS]	17
1.12. DATA AS A SERVICE [DAAS]	18
1.12.1 Security as a Service [SECaaS]	19
1.13. CLOUD COMPUTING AND WEB 2.0/WEB 3.0 INITIATIVES	20
1.14. IMPLIED BENEFITS OF CLOUD COMPUTING	23
1.15. DIFFICULTIES FACING CLOUD COMPUTING	25
1.15.1 Challenges according to a Provider Perspective	25
1.15.2. Difficulties according to a Customer Perspective	26
1.15.3 A Note on Security, Interoperability, and Portability Concerns	28
1.16. FINISHING UP REMARKS	29
REFERENCES	31

CHAPTER 2: VIRTUALIZAION IN CLOUD COMPUTING: EXISTING SOLUTION AND NEW APPROACHES

2.1. INTRODUCTION	34
2.1.1 Virtualization	34
2.2. IMPORTANCE OF VIRTUALIZATION	35
2.3. BENEFITS OF VIRTUALIZATION	36
2.4. TYPES OF VIRTUAL MACHINES	37
2.5. VIRTUAL MACHINE APPLICATIONS	37
2.6. VIRTUALIZATION COMPONENTS	38
2.7. VIRTUALIZATION TYPES	39
2.7.1. Full Virtualization	39
2.7.2. Para Virtualization	40
2.7.3. Hardware-Assisted Virtualization	41
2.8 VIRTUALIZATION SYSTEM	42
2.8.1. Xen Hypervisor	42
2.8.2. KVM Hypervisor	43
2.8.3 OpenStack	44
2.8.4 Storage	45
2.8.5 Server Virtualization	45
2.9 LIVE VIRTUAL MACHINE MIGRATION	46
2.9.1 QEMU and KVM	47
2.9.2 Libvirt	48
2.10 CONCLUSION	49
REFERENCES	50

CHAPTER 3: SECURITY ISSUES IN CLOUD COMPUTING AT THE VIRTUALIZATION LAYER

3.1. SECURITY TECHNIQUE FOR CLOUD COMPUTING ON THE VIRTUAL SERVER	52
3.1.1. Multi-tenant Situation	53
3.1.2. Control Loss	54
3.2. ISSUES WITH SECURITY IN SERVICE DELIVERY SYSTEMS	54
3.2.1. Security at SaaS	54

3.2.2. Security at PaaS	55
3.2.2. Security at IaaS	55
3.3. SECURITY IN VIRTUALIZATION	55
3.3.1. Security for VM	56
3.4. ISSUES WITH SECURITY DURING VM MIGRATION	57
3.5. SECURITY FOR HYPERVISORS	57
3.5.1. Attack from a VM to a VMM	57
3.5.2. Rootkits at the VMM level	57
3.6. SECURITY OF THE HOST MACHINE	58
3.6.1 Attack from a Guest to a Host	58
3.6.2 Rootkits at the kernel level	58
3.7. DoS	58
3.8. SURVEY	59
3.8.1. Loss of Control and Multi-Tenancy	59
3.8.2. Layers of Service Delivery Safety	60
3.9. SECURITY IN VIRTUALIZATION	61
3.10. SECURITY FOR VM	61
3.10.1. Outsider Invasion	61
3.10.2 Inter-VM Assault	63
3.11. ISSUES WITH SECURITY DURING VM MIGRATION	64
3.11.1. Hypervisor Security	65
3.12. HOST MACHINE SECURITY	66
3.12.1 Guest-to-Host Attack	66
3.13. DENIAL OF SERVICE ATTACK	67
3.13.1 Security model/architecture at Virtualization Layer	67
3.14 SUMMARY	68
REFERENCES	68

CHAPTER 4: SOFTWARE DEFINED NETWORK PROCESS FOR OVERCOMING ATTACKS IN VIRTUALIZATION

4.1. SERVICE ATTACKS	73
4.2. SDN-BASED DETECTION OF SERVICE ATTACKS	74
4.3. DDOS FLOODING ATTACKS ON THE NETWORK	74
4.3.1. DDoS flooding attacks at the application level	74

4.4. DEALINGS WITH A NEW BREED OF FIRMNESS AND AMENITY	75
4.4.1. Peer group for transportation	75
4.5. DATA COLLECTION ON TRAFFIC	77
4.6. THE CLOUD EDGE TO CORE MODEL	77
4.7. RESULTA OF THE EVUALTION	80
4.7.1. Extraction of features	81
4.7.2 Concern Outcome Estimation	83
4.8. SUMMARY	84
REFERENCES	84

CHAPTER 5: BUILDING AN INSTRUCTION DETECTION SYSTEM USING MACHINE AND DEEP LEARNING APPROACHES IN VIRTUAL CLOUD SYSTEM

5.1. INTRODUCTION	86
5.2. BACKGROUND OF DNN	87
5.2.1. Extraction of features	87
5.3. OPTIMIZATION STRATEGIES FOR GENETIC ALGORITHM	88
5.3.1. Parallel processing	88
5.3.2. Fitness Value Hashing	89
5.4. SIMULATED ANNEALING ALGORITHM	91
5.5. THE PROPOSED SYSTEM	91
5.5.1 Approach of our proposed system	91
5.6. ROLE OF SIMULATED ANNEALING ALGORITHM IN THE PROPOSED SYSTEM	95
5.7. FRAMEWORK OF OUR PROPOSED MLIDS	96
5.8. POSITIONS OF THE PROPOSED SYSTEM IN A CLOUD NETWORK	97
5.9. EXPERIMENTATION	99
5.9.1. Data preprocessing	99
5.9.2. Machine learning optimization framework IGASAA	100
5.9.3. Experimentation based on CICIDS 2017 dataset	100
5.9.4. Experimental results	102
5.10. EXPERIMENTATION BASED ON NSL-KDD DATASET	103
5.10.1. Description of NSL-KDD dataset	103
5.10.2. Experimental results	104
5.10.3. Experimentation based on CIDDS-001 dataset	104

5.11. CONCLUSIONS AND FUTURE WORK	106
REFERENCES	106

CHAPTER 6: CASE STUDY OF REAL TIME APPLICATION SET-UP IN CLOUD COMPUTING

6.1. INTRODUCTION	110
6.1.1. Kernel-Based Virtual Machine	110
6.1.2. Xen	111
6.1.3. Secure Data Analysis in GIS	111
6.1.4. Database	111
6.1.5. Data Mining and Techniques	112
6.1.6. Distributed Database	112
6.1.7. Spatial Data Mining	112
6.1.8. Secure Multi-Party	112
6.1.9. Association Rule Mining Problem	113
6.1.10. Distributed Association Ruling	114
6.1.11. Kernel-Based Virtual Machine	115
6.1.12. Xen	115
6.2. GREEN COMPUTING'S ASCENSION IN THE MODERN COMPUTING ENVIRONMENT	115
6.3. GREEN COMPUTING	120
6.4. SUMMARY	121
REFERENCES	121

PREFACE

Cloud computing isn't a new concept, nor is it particularly difficult in terms of internet technology. What's new is the maturation and expansion of cloud computing methodologies and tactics that help companies achieve their business agility goals. In retrospect, the word "utility computing" did not capture or create the same buzz in the information business as "cloud computing" has in subsequent years. Nonetheless, there is a growing desire for easily available resources, and the pragmatic or serving aspects lie at the heart of freelancing access to data technology services and resources.

As businesses and information systems executives recognize the value of merging and distributing computer resources rather than developing and sustaining them, big data has become a major changer in the industry. There appears to be no shortage of opinions about the advantages of cloud computing, nor of companies eager to provide services in either open source or offering commercial solutions. Beyond the hoopla, several elements of the Cloud have gained fresh attention as a result of their enhanced service capabilities and potential efficiency.

Dac Nhuong Le et al. take the industry beyond simple definitions of virtualization and cloud computing, network and deployment methods, and compare them in day-to-day operations in Virtualization and Cloud computing. Dac-Nhuong Le and his co-authors guide the reader through the fundamental components of cloud computing, including its history, development, and needs, from beginning to conclusion.

They clarify service needs, infrastructure, privacy, and outsourcing of key computer resources through studies and architectural models. The use of virtualization in data centers necessitates the development of a new class of networks to enable resource flexibility, increased mobile workloads, and the transition to virtual load production, all of which necessitate optimum availability. Building a network with continuous functionality that spans both data centers and virtual servers necessitates a new architectural strategy for IT system design and construction. The administration of the physical and virtual network connection, and also performance, flexibility, and logical addressing patterns, must all be considered. A virtualization-ready network, once installed, may provide a plethora of innovative services via a single shared infrastructure.

VMware, Citrix, and Microsoft virtualization techniques wrap and isolate current programs from computer machines. Virtual machines, in contrast to real machines, are defined by a portable software image that may be created on physical hardware at any time. Virtualization provides flexibility, allowing computer capacity to be scaled up or down on demand by changing the number of virtual machines running on a particular physical server. Virtual machines can also be moved from one physical server to another while still in use. To take it a step further, virtualization provides “location freedom,” allowing virtual machines to be transported over ever-greater distances. As cloud platforms and multi-tenancy capabilities evolve and mature, aggregating resources across apps, business divisions, and distinct companies to a single shared, yet segmented, infrastructure can achieve economies of scale.

CHAPTER 1

CLOUD COMPUTING: A COMPREHENSIVE INTRODUCTION

1.1 INTRODUCTION

The process of providing computing and communications related services with the guide of distantly found, network-based resources without a client of such resources claiming these resources is popularly known as the cloud computing. The organization being referred to regularly, however not really, is the Internet. The resources provisioned incorporate a scope of administrations including information, programming, stockpiling, security, etc. For instance, for a mail administration like Gmail, watch a film on YouTube, shop at Amazon.com, or store records utilizing DropBox, cloud-based resources [1] will be utilized. In this part, the development of Cloud Computing from its initial roots in centralized server based registering to the current day was discussed and furthermore clarify the various administrations delivered by Cloud Computing in the present business and individualized computing settings. This section gives a complete perspective on the quickly thriving field of Cloud Computing and makes way for additional top to bottom conversations on its security, trust, and administrative angles somewhere else in this abstract.

1.2 THE EVOLUTION OF CLOUD COMPUTING

The descriptive word "Cloud" in Cloud Computing alludes to the organization utilized for administration provisioning. In graphs portraying cloud-based administrations, the cloud is regularly in a real sense portrayed as the diagram of a hand-drawn cloud on paper. The utilization of cloud-like shapes in outlines portraying organizations, for example, the Internet goes back numerous years and is a staple of standard course readings and articles on information correspondence organizations. The expression "Distributed computing," however, is moderately new. To more readily grasp this somewhat early wonder, history and inspect prior models of provisioning administrations over a correspondences organization, i.e., the forerunners of present-day Cloud Computing are considered.

1.2.1 Time-Sharing on Mainframe Computers

The mid 1950s saw the approach of business "centralized server" PCs, for example, the IBM 701. These PCs were single-client, non-shareable, each work in turn frameworks and were leased by organizations for about \$25,000 per month. A few software engineers joined, on first-start things out served premise, for "meetings" on a centralized computer where every meeting was a square of time committed to preparing a solitary "work" [i.e., a program]. Every software engineer required around 5 minutes to set-up his/her work incorporating punching in at a mechanical clock, hanging an attractive tape, stacking a punched card deck, and squeezing a "heap" catch to start work handling [2.].

Failures in the process because of exorbitant manual intercession brought about much burned through handling time even as occupations were lined and regularly deferred. To further develop measure effectiveness, General Motors [GM] and North American Aviation [NAA] and a piece of Boeing fostered a working framework, the GM NAA I/O [Input/Output] framework and put it into creation in 1956 [2]. This proclaimed the coming of "clump handling" where different positions could be set up immediately and to culmination without manual mediation [3].

Further enhancements were acknowledged with the coming of the IBM System 360 centralized computer in 1964 what isolated I/O undertakings from the CPU [Central Processing Unit] and cultivated these out to an I/O sub-framework, in this manner opening up the CPU to perform calculations needed by a second occupation when another work was hindered for I/O tasks. Group preparing offered a few advantages: Individual positions in a bunch could be handled at various occasions dependent on asset accessibility, framework inactive time was decreased, framework usage rates were improved and, as an outcome, per-work handling costs were diminished.

With cluster preparing, a PC's time is viewed as impressively more important than a human's and human work is planned around the machine's accessibility. Conversely, "intelligent registering," considers a human's time similar to the more important and perspectives a PC just as a fit "associate." Early executions of intuitive figuring incorporate the IBM 601 that permitted a solitary client intuitive use at a time. Nonetheless, permitting one client to consume a scant asset additionally brought about significant shortcoming in asset usage. Then again, offering a

few intelligent clients apparently simultaneous use would bring about better utilization of the electronic right hand [4].

In 1961 MIT presented the world's first Time Sharing Operating System, the Compatible Time Sharing System [CTSS]. At the appointed time, IBM presented a Time Sharing Option [TSO] in the OS 360 working framework utilized in the IBM System 360. Time Sharing presented further handling efficiencies over cluster preparing. Maybe than measure a task completely, time sharing would commit a brief term of time called a "period cut" to preparing a task and afterward directs to give comparative concentration toward another work. The CPU quickly changes from one occupation to another that it appears to every client that his/her work have the full and complete consideration of the CPU - a client encounters no recognizable postponements.

A characteristic outgrowth of intuitive processing was far off admittance to a PC by means of terminals. A few terminals were "multiplexed" over phone lines utilizing singular modems to interface clients to a solitary centralized server. Shared centralized server intuitive access and use by means of multiplexed terminals and the phone organization might be respected the soonest Cloud Computing model despite the fact that it was then alluded to as Time-Sharing. During the 1960s, a few merchants offered Time-Sharing "administrations" to organizations. These included Tymshare, National CSS, Dial Data, and BBN utilizing hardware [centralized computer and minicomputers] from IBM, DEC, HP, CDC, Univac, Burroughs, and others.

1.2.2 Peer-to-Peer and Client-Server Computing

The approach of industrially feasible [PCs] from Apple, Commodore, and Tandy Corp., combined with the rising utilization of PCs starting in the last part of the 70s and well into the 80s and past proclaimed the decay of Time-Sharing with bigger [centralized server and little] PCs [5]. At first, PCs, aside from use for business figuring utilizing programming like VisiCalc [accounting page] and Lazy Writer [word handling], were likewise utilized as terminals to associate with the bigger machines by running terminal imitating programming. Before long the market opened up to merchants like Atari, TI, NEC and others. In 1981, IBM entered the quarrel with its IBM PC. At the appointed time, numerous clients found that the consolidated preparing

capacity of various arranged PCs was adequate to address their issues. Such "mists" started multiplying in two structures – shared and customer worker processing.

1.3 THE PEER-TO-PEER MODEL

In shared [or P2P] processing, every PC in the organization can go about as both a help requestor and a specialist co-op for the leftover PCs in the organization/cloud. This worked with the sharing of costly and additionally scant resources, for example, information records, printers, scanners, hard circles, and tape drives. There is no focal power or "expert," like a centralized server in time-sharing, where every terminal acted in a docile, "slave" job and just when the centralized server made time accessible for it. In a P2P organization, each PC could go about as expert or slave at various ages. P2P networks empowered intra-and between authoritative organizations with each extra PC added to the organization carrying added capacity to the framework.

Simultaneously, the organization was stronger than one with an expert slave plan as it didn't have the weakness of a weak link – in the event that one or a couple of hubs [i.e., PCs] in a P2P network were to fizzle, the remainder of the "cloud" could keep working. The Internet, initially considered as the ARPANET in the last part of the 1960s by the US Department of Defense, was a P2P framework [6]. Its objective was to work with the sharing of figuring resources around the U.S. utilizing a typical organization engineering that would permit each host to be an equivalent player. While early, broadly utilized applications like Telnet and FTP were Client-Server applications, the framework overall was P2P as each Internet host could Telnet or FTP some other host and has were not related in ace slave connections. The inescapable sending of PCs, first in quite a while and afterward in homes, powered the quick expansion of P2P processing during the 80s and after.

1.3.1 The Client-Server Model

PCs were additionally taken advantage of in an alternate way which might be viewed as a media between the totally imperious centralized computer moronic terminal model and the completely fair P2P model. The Client-Server model was presented by Xerox PARC [Palo Alto Research Center] during the 70s. This model allocates one of two jobs, in particular customer or worker to

each PC on an organization. In this manner, a solitary PC may not go about as customer or worker like in the P2P model. Simultaneously, the organization isn't limited to facilitating different, frail customers fastened to a solitary, incredible, worker as in centralized computer terminal organizations. A worker makes accessible resources in its domain that a customer looks for.

Every worker additionally manages different customer demands and does as such in a period shared way as talked about before with centralized server figuring. The Internet, what began as to a great extent a P2P organization, transformed after some time into a generally customer worker organization. This progress was sped up by the Internet blast of the mid 90s when the overall population, and not simply researchers, rushed to the net as a method for email trades, web perusing, and web based shopping. This change has proceeded until the post-year 1998 reappearance of P2P applications like Napster, Gnutella, Kazaa, and BitTorrent for music, film, and game document sharing. Signs are that the Internet will probably proceed as a half breed climate facilitating both P2P and Client-Server applications for a long time to come.

1.4 GRID COMPUTING

One more worldview in Cloud Computing's development is Grid Computing starting points date back to the 90s. This worldview was spurred by similarity to the electrical force matrix that gives inescapable, trustworthy, and predictable admittance to utility force [7]. Lattice Computing is a variation of Cluster Computing with the distinction being that the PCs on a framework could be heterogeneous, approximately coupled/powerfully saddled, and geologically scattered frameworks. Lattice Computing likewise varies from traditional Distributed Computing by underlining enormous scope asset sharing, dedication to imaginative applications, and elite. A Grid Computing framework is likewise a self-ruling framework in that it intends to act naturally arranging, self-tuning, and self-mending.

Together the individuals from a framework structure a "virtual" super-PC whose force and assets are accessible to a client dependent on asset accessibility, ability, execution, cost, and nature of administration assumptions. Lattice Computing frameworks are given to handling convoluted assignments like the quest for extra-earthbound knowledge [SETI], protein collapsing, drug

plan, sub-atomic displaying, monetary demonstrating, high-energy physical science, mind movement examination, quake and fighting reenactment, and environment/climate demonstrating [8]. A particularly huge Grid Computing framework is the WLCG [Worldwide LHC Computing Grid] crossing in excess of 170 figuring communities in 36 nations. Its motivation is to store, convey, and investigate the almost 25 petabytes of information created each year by the Large Hadron Collider [LHC] at CERN [Conseil Europeen pour la Recherche Nucleaire [French], or the European Council for Nuclear Research], Geneva [9].

1.5 UTILITY COMPUTING

The Utility Computing model was likewise roused by utility administrations like power yet in a way not the same as Grid Computing. Utility Computing drew its motivation from how open utility clients pay metered rates dependent on utilization. In comparable soul, Utility Computing tries to make accessible cloud-based processing assets to clients for installment dependent on utilization. Fundamentally, a client reevaluates all or part of its registering asset needs to another substance, the Utility Computing administrations supplier [10].

The key distinctive trademark is that the expense for administrations is definitely not a level charge yet is utilization based. The advantages to a client are that it is vindicated of claiming such assets and of orderly obligations like those identified with obtaining, lodging, establishment, upkeep, investigating, redesigning, getting, and utilization. However the expression "Utility Computing" was first hypothesized in quite a while, as IBM with their centralized computer time-sharing models were, it could be said, pioneers in Utility Computing [11].

Following the decrease in centralized server interest and the development of PCs, Utility Computing reemerged in the last part of the 1990s and mid 2000s with merchants like InSynQ, HP, Sun Microsystems and Alexa building up Utility Computing administrations. Today, Utility Computing merchants incorporate very much perceived names like IBM, Amazon, and Google. Of the different Cloud Computing-related chronicled achievements examined hitherto, the Utility Computing worldview is maybe nearest in soul to introduce day Cloud Computing and

furthermore the reason for much disarray with respect to what recognizes it from Cloud Computing.

1.6 VIRTUALIZATION

Another idea basic Cloud Computing is Virtualization. Virtualization alludes to the reenacted making of something – a PC, a working framework, a capacity gadget, or some other registering or correspondence asset without having a physical/real occurrence of it. This idea goes back numerous many years and was spearheaded starting in the mid 1960's by substances like GE, IBM, MIT, and Bell Labs. Following a couple of long stretches of exploring different avenues regarding one-off, research facility forms of the idea, the IBM CP-67 centralized server, dispatched in 1968 and running the CP-CMS working framework, was the primary business PC to help Virtualization and was introduced at eight client locales [12]. There are a few sorts of Virtualization in the figuring scene.

A conversation incorporating these sorts is past the current extension. Equipment or Platform Virtualization is a typical case that portray straightaway. As a general rule, Hardware Virtualization brings about the formation of at least one "visitor" or "virtual" machines [VM] running inside a "have" or "genuine" machine. This might be cultivated with the guide of programming commonly called a Hypervisor or Virtual Machine Monitor [VMM]. Instances of VMMs incorporate Microsoft's Virtual Server, VMWare's GSX, and IBM's VM/ESA. To every one of numerous visitor clients upheld by a solitary host, maybe a disengaged, independent PC is accessible for his/her utilization albeit each of these is a virtual machine and not a real/actual PC.

The degree of virtualization in Hardware Virtualization could likewise vary. There are three degrees of Hardware Virtualization called Full Virtualization [close total equipment climate reenactment to permit visitor applications to run un-changed], Partial Virtualization [some equipment climate components, however not all, are mimicked allowing a few applications to run un-adjusted], and Para Virtualization [definitely no equipment climate reproduction except for visitor applications run in disengaged spaces and should be altered]. Two normal types of Hardware Virtualization are Server Virtualization and Desktop Virtualization. Along these lines

a solitary, physical [i.e., have] worker could uphold different virtual workers, bringing about less actual worker occurrences, energy reserve funds, and support ease.

Work area Virtualization [likewise called Desktop as a Service [DTaaS], Virtual Desktop, or Hosted Desktop Services] permits clients to get to a whole registering climate through a distant customer gadget, for example, a cell phone, tablet, or PC by running work areas as Virtual Machines on a supplier's worker where all work area client conditions are overseen and gotten. By re-appropriating Desktop Virtualization, issues, for example, asset provisioning, load adjusting, organizing, back-end information stockpiling, reinforcement, security, and redesigns are taken care of by DTaaS suppliers like Citrix. Note that Hardware Virtualization is unmistakable from Time Sharing. Customary Time Sharing gives a whole host PC to various clients yet at various occasions – like at similar, independent machines, all simultaneously accessible to numerous clients.

With Virtualization [Beal, 2012], there is the potential for more effective utilization of assets [i.e., less actual machines, better space productivity, better energy proficiency [decreased electrical utilization and diminished cooling costs], better security [every client could be running a different working framework and not sharing one], and expanded unwavering quality [a solitary client couldn't crash the whole framework, just his/her Virtual Machine]. Nonetheless, Virtualization likewise claims a cost – the more VMs that are conveyed, the more noteworthy the expected corruption in execution of each VM there still are protection and security dangers to the numerous clients sharing a solitary actual host. All things considered, as seen along this part, Virtualization is being saddled as a key empowering agent of current Cloud Computing.

1.7 SERVICE-ARRANGED ARCHITECTURE

A further advancement that underlies Cloud Computing is the Service-arranged Architecture [SOA]. SOA permits an application's business rationale or individual capacities to be modularized and introduced as administrations for other customer/customer applications [Kodali, 2005]. These help modules are "approximately coupled" as in the assistance interface of a module is autonomous of its execution. Accordingly, application designers or framework

integrators can assemble applications by drawing upon administration modules depending on the situation regardless of their hidden execution subtleties.

For example, a help can be carried out one or the other in .Net or J2EE [Java 2 Platform Enterprise Edition], and the application burning-through the assistance could utilize an alternate stage or language. Every module offers a little scope of basic administrations to different segments. The modules can be consolidated and re-joined from multiple points of view and contain a profoundly adaptable data innovation [IT] applications framework. In this manner, SOA is a way to deal with building IT frameworks that permits a business to use existing resources, make new ones, and effectively empower changes that organizations definitely should oblige over the long run [13].

Note that the accentuation of SOA is on programming reusability – don't "discard" exertion previously put in or burn through energy re-designing the wheel similarly as with coding every application without any preparation. With the accentuation on re-utilizing straightforward code modules, an organization doesn't encounter the customary turn of events and upkeep time and cost disadvantages ordinarily connected with an expansion of costly, solid frameworks drawing on heterogeneous innovation. Every product module in SOA is something that gives a business administration and these business administrations are perpetually shared by various applications broad. Through programming repetition decrease or disposal, SOA gives a significant part of similar advantages related with programming consistency, viability, and adaptability [14].

1.8 THE CLOUD COMPUTING PARADIGM

The above-examined registering standards, while unmistakable from each other in certain regards, additionally have shared characteristics. An overwhelming normal trademark is that every one of them includes provisioning shared assets over an organized framework called as a "cloud." Today, however, there is impressive uproar over the latest worldview in this developmental chain, called Cloud Computing. What is Cloud Computing and how can it contrast, if by any means, from any or the entirety of the ideal models just examined? There is significant disarray encompassing the term. This disarray isn't limited just to the layman or

forthcoming customers of the help. At Oracle OpenWorld 2008, Oracle Corp. Chief Larry Ellison broadly noted [15]:

"The intriguing thing about Cloud Computing is that we've re-imagined Cloud Computing to incorporate all that which has been done now. I can't consider anything that isn't Cloud Computing with these declarations.

The processing local area everywhere was [and, maybe, is] separated ... about half accepted that Ellison was totally right in the evaluation and the other viewed him as an apostate. As far as this is concerned, Ellison was not contending that Cloud Computing was a prevailing fashion but rather that the mark was a trend that everyone had been rehearsing Cloud Computing in some structure for various years without utilizing the name Cloud Computing. All the more as of late, Ellison finished his subsequent perception and, as of late as September 2012, reported the dispatch of an Infrastructure-as-a-Service [IaaS] cloud administration [16], one of a few potential administrations discussed in this Section 4. Looking at accessible writing on Cloud Computing loans trustworthiness to Ellison's perspective as numerous clarifications of Cloud Computing miss the mark. As one model [Biswas, 2011]:

"Actually like water from the tap in your kitchen, Cloud Computing administrations can be turned on or off rapidly depending on the situation. Like at the water organization, there is a group of committed experts ensuring the help gave is protected, secure and accessible on a day in and day out premise. At the point when the tap isn't on, in addition to the fact that you are saving water, however you're not paying for assets you don't at present need." This clarification makes one wonder, "What, then, at that point, is Utility Computing? " and adds to the disarray in a generally confounded customer base. Then again, there are different depictions that better lucid what Cloud Computing is and why it is unique.

The National Institute of Standards and Technology [NIST] characterizes Cloud Computing in this way [17]. Distributed computing is a model for empowering omnipresent, advantageous, on-request network admittance to a common pool of configurable figuring assets like the networks, workers, stockpiling, applications, and administrations that can be quickly provisioned and delivered with insignificant administration exertion or specialist organization association. The

vital knowledge on how Cloud Computing varies from its archetypes is contained in the last piece of this definition have stressed for accentuation. Perry [2008] explains on this perspective:

"The large news is for application engineers and IT activities. Done right, Cloud Computing permits them to create, send and run applications that can undoubtedly develop limit [adaptability], work quick [execution], and never — or possibly seldom — come up short [dependability], all with no worry concerning the nature and area of the basic framework. So despite the fact that they are frequently lumped together, the contrasts between Utility Computing and Cloud Computing are essential. Utility Computing identifies with the plan of action where application framework assets — equipment or potentially programming — are conveyed. While Cloud Computing identifies with the manner in which the configuration, assemble, send and run applications have been working in a virtualized climate, sharing assets and flaunting the capacity to powerfully develop, psychologist and self-recuperate."

It is seen that while the qualification between Cloud Computing and its archetypes are questionable from a purchaser viewpoint, there is an unmistakable differentiation according to a supplier viewpoint. The focal point of Cloud Computing is on alleviating or killing issues related with customary application advancement and opening up hierarchical IT units to zero in on business technique and how to best use cloud-based IT to help that methodology. As ought to become clear in our conversations that follow, Cloud Computing, contingent upon needs, endless supply of the prior ideal models like Utility and Grid Computing, Client-Server and Peer-to-Peer Computing, Virtualization, and Service-Oriented Architecture. These previous ideal models include the structure squares of present-day Cloud Computing.

1.9. DISTRIBUTED COMPUTING MODELS

Until this point, four models for Cloud Computing administrations organization have been characterized and that are Private Cloud, Community Cloud, Public Cloud, and Hybrid Cloud [19] as discussed below.

1.9.1 Private Cloud

A Private Cloud [InfoWorld, n.d.] is cloud framework set up for sole use by a solitary association. It is additionally alluded to as an Internal Cloud or Corporate Cloud [Rouse, 2009a].

Either the association's corporate organization or server farm directors become cloud specialist co-ops working behind the corporate firewall or the Private Cloud is facilitated by a third-gathering supplier exclusively for the association. In any case, such a cloud serves the association's "inside" clients and nobody else. Whenever built in-house, setting up a private cloud requires a huge obligation to virtualizing the business climate. Generally, the association should obtain, fabricate, and oversee such a cloud all of which calls for significant asset speculations.

A reevaluated Private Cloud likewise is more costly than a Public Cloud. In this way, a Private Cloud normally invalidates the expense benefits that build with reevaluating IT foundation which is an essential objective of relocating to the cloud for some associations. However a few associations have looked to set up Private Clouds for reasons other than cost reserve funds, like better control, security, protection, customization adaptability, and unwavering quality, versus a Public Cloud administration [20]. Private Cloud toolboxes incorporate IBM's Web Sphere Cloudburst Appliance, HPs Cloud Start, and Amazon's Virtual Private Cloud.

Outsider Private Cloud merchants, for example, Saber Holdings set up Private Clouds for singular carriers and Siemens has set up individual secure, virtual test habitats for its accomplice associations. Different instances of Private Cloud execution incorporate NASA's Nebula and the US Defense Information Systems Agency's Rapid Access Computing Environment [RACE] program [21]. As is apparent from these models, Private Clouds will in general be sent by enormous associations with critical assets available to them.

1.9.2 Public cloud

A Public Cloud [22] is one that is worked and overseen at datacenters having a place with specialist co-ops and shared by numerous clients [multi-occupancy]. All things considered, it doesn't live behind an association's corporate firewall nor is it implied for elite use. A new Trend Micro study [23] of 1200 associations with no less than 500 workers in 6 nations noticed that 93% demonstrated that they were utilizing no less than one cloud specialist co-op, 38% felt that their cloud specialist organizations were neglecting to meet their IT and business needs, 43% had a security slip by somewhat recently, 55% were concerned.

1.10. DISTRIBUTED COMPUTING SERVICES

Given our comprehension of the expression "Distributed computing," its advancement, and models for organization, the inspection on the various administrations delivered by Cloud Computing were done based on client needs. Figure 1 pictorially portrays the different Cloud Computing administrations and supplier and client connections with the cloud and its administrations utilizing wired and remote gadgets. These administrations length a customer's framework needs, application needs, security needs, stockpiling needs, etc.

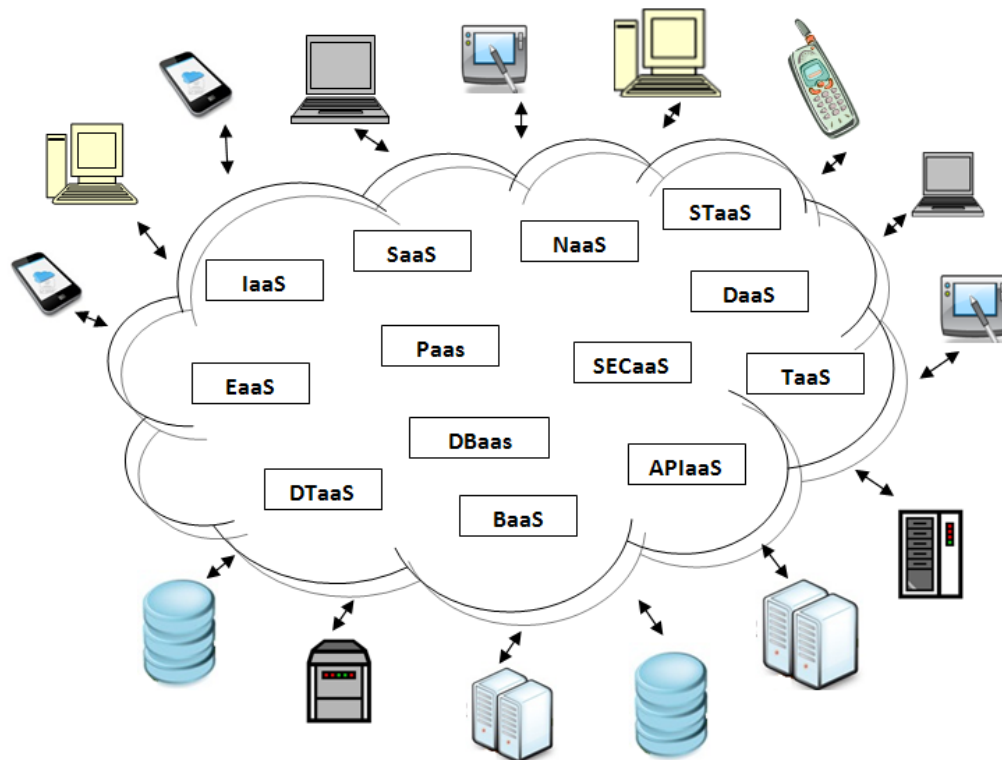


Figure 1. Shows cloud computing's conceptual view

As per the NIST, Infrastructure as a Service [IaaS], along with Platform as a Service [PaaS] and Software as a Service [SaaS], are the three central Cloud Computing administration models [28]. These three assistance models follow a PC's engineering and offer types of assistance at the equipment, framework, and application level individually [29]. Here, the three fundamental variations and sub-classes are discussed inside each in the accompanying areas just as a couple

of other assistance models in particular, Data as a Service [DaaS] and Security as a Service [SECaaS].

1.10.1 Infrastructure as a Service [IaaS]

IaaS, an equipment level help, gives registering assets like preparing power, memory, stockpiling, and organizations for cloud clients to run their applications on-request [Stallings and Case, 2013]. This permits clients to amplify the usage of figuring limits without purchasing and deal with their own assets. It addresses a change in perspective from review foundation as a resource for seeing it as a re-appropriated administration. IaaS suppliers [e.g., Amazon EC2, Windows Azure Virtual Machines, Google Compute Engine] have PCs as Virtual Machines that are overseen by low-level codes called hypervisors, for example, Xen or KVM to meet clients' processing needs. IaaS clients pay for assets designated and burned-through on a Utility Computing premise and partake in the adaptability of progressively increasing their figuring foundation or down as per asset requests without bringing about capital consumptions on these assets that are frequently underutilized [30].

a. Network as a Service [NaaS]

Organization as a Service [NaaS], an occurrence of IaaS, gives clients required information correspondence ability to oblige rushes in information traffic during information serious exercises, for example, video conferencing or enormous record downloads. NaaS suppliers [e.g., Verizon, AT&T] work utilizing three normal help models: virtual private organization [VPN], transmission capacity on request [BoD], and portable virtual organization [MVN]. VPN broadens a private organization's usefulness and strategies across open organizations like the Internet.

Body progressively dispenses transfer speed to bursty traffic requests by different clients. MVN is a versatile organization that isn't claimed by a portable administrations supplier however is rented from a substance that possesses the foundation. Fundamentally, the lessor gives NaaS to the tenant, who thus offers required types of assistance to end customers [i.e., is an affiliate]. By considering systems administration and registering assets in general, NaaS suppliers are better ready to upgrade these asset portions to clients with network availability administrations.

b. Storage as a Service [STaaS]

Capacity as a Service [STaaS], a type of IaaS, gives stockpiling framework on a membership premise to clients who need a minimal expense and advantageous approach to store information, synchronize information across numerous devices, oversee offsite reinforcements, alleviate dangers of catastrophe recuperation, and protect records as long as possible. With information developing at a yearly pace of more than 20%, capacity limit prerequisites should be multiplied each a few years. As per InformationWeek Analytics Public Cloud Storage Survey, filing messages and meeting maintenance arrangements were the top purposes behind capacity development.

By re-appropriating stockpiling to STaaS suppliers [e.g., Amazon Simple Storage Service, IBM Tivoli Storage Manager], clients shift the weight of limit the board, activities, and support to the supplier. Likewise, the developing utilization of cell phones, for example, cell phones, PCs, and tablets to get to organization information will strengthen the requirement for information solidification and synchronization. STaaS clients approach their information whenever anyplace over the Internet. They can indicate how frequently and what information ought to be supported up.

They can likewise demand a duplicate of the information if there should create an occurrence of information debasement or misfortune. Surely, reinforcement, debacle recuperation, record chronicling and email documenting were the top reasons that associations were thinking about distributed storage. In any case, worries over security, protection, unwavering quality, accessibility, execution, information misfortune and administration interruptions, particularly for enterprises like medical care, financials, and legitimate administrations, may keep clients from utilizing STaaS to store their essential [or crucial] information.

c. Database as a Service [DBaaS]

Information base as a Service [DBaaS], likewise identified with IaaS, gives clients consistent components to make, store, and access data sets at a host site on request. DBaaS suppliers are likewise liable for the administration of the whole information base including reinforcement, organization, reclamation, rearrangement, and movement. Cloud-based information base

frameworks, for example, Google BigTable, Amazon Simple DB, and Apache HBase permit clients to submit questions to data sets with nonexclusive compositions. Google Cloud SQL permits clients to make, design, and utilize social data sets inside Google App Engine applications. Information protection and security stay the vital worries with DBaaS.

d. Backend as a Service [BaaS]

Backend as a Service [BaaS], a kind of IaaS, gives web and versatile application engineers an approach to associate their applications to backend distributed storage with added administrations like client the board, pop-up messages, informal organization administrations combination utilizing custom programming advancement packs and application programming interfaces. BaaS clients save time and exertion in having a predictable method to oversee backend information and administrations

e. Desktop as a Service [DTaaS]

Another generally utilized case of an infrastructural administration is Desktop as a Service [DTaaS] or Desktop Virtualization to a concise depiction that shows up in Section 1.3 on Virtualization.

1.11 PLATFORMS AS A SERVICE [PaaS]

Platform as a Service [PaaS], a framework level help, gives clients a processing stage for the turn of events, testing, organization, and the executives of uses. Clients fabricate their own applications utilizing programming dialects and devices upheld by PaaS suppliers. These applications then, at that point run on a supplier's foundation and additionally are conveyed to end-clients through the Internet from the supplier's workers. PaaS suppliers [e.g., Amazon Elastic Beanstalk, Windows Azure Compute, and Google App Engine] unite middleware like data sets, working frameworks, and devices to help programming advancement and conveyance on compensation. PaaS clients acquire proficiency and efficiency with a normalized application advancement measure without the expense and intricacy of apportioning PC and capacity assets to coordinate with interest for improvement and testing exercises

1.11.1 Software as a Service [SaaS]

Programming as a Service [SaaS], an application level help, permits clients to get to supplier's applications from different gadgets through a flimsy customer interface like an internet browser. SaaS suppliers [the most popular model is salesforce.com] utilize a multi-inhabitant engineering to convey a solitary application to a great many clients on a membership premise. SaaS clients' access required applications without the problems related with programming permitting, establishment, support, updates, and applying patches. SaaS is particularly interesting to private companies that can't stand to claim and oversee top of the line undertaking programming like bookkeeping, invoicing, human asset the board, client relationship the executives, and endeavor assets arranging. What's more, SaaS work area applications [e.g., Google Apps, Office 365, Zoho Office] give cooperative prospects that permit clients from far off areas to cooperate on a similar application record continuously through the Web.

a. Testing as a Service [TaaS]

Testing as a Service [TaaS], furnishes clients with programming testing capacities like age of test information, age of experiments, execution of experiments, and test outcome assessment on a compensation for each utilization premise [Yu et al., 2010]. Programming testing requires exorbitant and broad registering assets like workers, stockpiling, and organization gadgets yet for a restricted time frame. Thus it's a good idea to re-appropriate testing assignments to TaaS suppliers. For instance, UTest offers utilitarian, security, burden, limitation, and ease of use testing administrations.

b. API as a Service [APIaaS]

Programming interface as a Service [APIaaS] is firmly identified with SaaS, however rather than conveying out and out applications as in SaaS, APIaaS gives Application Programming Interfaces [API] for clients to take advantage of usefulness of such Web administrations as Google Maps, finance handling [e.g., by ADP Inc.], and Mastercard preparing administrations [e.g., through Merchant Express]. APIaaS offered by PaaS suppliers, for example, Google App Engine permits designers to construct highlight rich applications to perform different functionalities including: application log keeping and getting to, huge informational collection

age and preparing [MapReduce], Secure Socket Layer scrambled applications, site execution examinations and enhancement [PageSpeed], area mindful inquiries, client verification, texting, moment update program channel foundation, application errands booking and execution, web content recovery , and language interpretation.

c. Email as a Service [EaaS]

Email as a Service [EaaS], a case of SaaS, gives clients a coordinated arrangement of messaging, office mechanization, records the executives, relocation, and incorporation administrations with chronicling, spam hindering, malware assurance, and consistence highlights. Messaging administrations incorporate calendaring, contacts and texting, cell phone joining, and search abilities. Office mechanization administrations incorporate web conferencing, report sharing, and program based office usefulness suites. Records the board administrations incorporate coordinating report the executives with email, and giving APIs to records looking and the executives. Movement administrations incorporate relocating email frameworks and information, end client preparing, and moving portable clients. Incorporation administrations incorporate turn of events and specialized support for combination of utilizations and undertaking the executives. EaaS clients receive the rewards of saving money on authorizing and keeping up with the equipment and programming of an organization's email framework ["Email as a Service [EaaS]", n.d.].

1.12 DATA AS A SERVICE [DAAS]

Information as a Service [DaaS] gives information on request to a different arrangement of clients, frameworks, or applications. Driving DaaS suppliers, for example, DataDirect offer programming to associate business applications to information whereby information network for conveyed frameworks is improved and smoothed out. Information encryption and working framework validation are ordinarily accommodated added security. DaaS clients approach great information in a unified spot and pay by volume or information type, depending on the situation.

Be that as it may, in light of the fact that the information is claimed by the suppliers, clients can just perform read procedure on the information. Regardless with the worldwide information volume arriving at 1.8 zettabytes [a zettabyte is around a trillion gigabytes] in 2011 and developing by a factor of nine of every five years, consideration has been moved to Big DaaS, for example, Google's Public Data administration that totals, oversees, and gives admittance to information on open foundations and government organizations. Google DaaS clients can utilize Google Public Data Explorer to pound up and envision information powerfully.

1.12.1 Security as a Service [SECaaS]

Security as a Service [SECaaS] is another way to deal with security wherein cloud security is moved into the actual cloud whereby cloud administration clients will be shielded from inside the cloud utilizing a bound together way to deal with dangers. Four components of cloud security are as of now gave: email separating, web content sifting, weakness the board, and personality the executives. Email administrations are secured where they dwell against malware, spam, and phishing dangers through email separating. Web content separating incorporates URL sifting, HTTP header screening, page content and inserted joins examinations, and active web traffic observing to obstruct touchy data like IDs, Visa data, and licensed innovation from being compromised.

Weakness the executives shields customers from a common climate utilizing application firewalls between virtual machines, virtual interruption identification frameworks, cloud antiviruses, and virtual private organizations connecting virtual machines. Personality the board incorporates recognizable proof solicitations to guarantee principles consistence, single sign-on interoperability, and arrangements of ID, verification, approval, and responsibility functionalities.

1.13 CLOUD COMPUTING AND WEB 2.0/WEB 3.0 INITIATIVES

Recently, the distinctive Cloud Computing administrations accessible to clients were depicted. As recently noticed, the NIST [12] characterizes Cloud Computing as, " ... a model for empowering universal, advantageous, on-request network admittance to a common pool of configurable processing assets [e.g., networks, workers, stockpiling, applications, and administrations] that can be quickly provisioned and delivered with insignificant administration exertion or specialist organization association." This perspective on Cloud Computing has brought about some disarray with another term that is at present extremely well known in registering circles, specifically, Web 2.0. Here, the characterization of Web 2.0 and endeavor to draw qualifications and connections between Cloud Computing and Web 2.0 is done.

The term Web 2.0 was first presented by [31] in the article, "Divided Future." Dinucci noticed then the start of another method of utilizing the web that she named Web 2.0 rather than its archetype, Web 1.0. Web 1.0 [23] is the first World Wide Web as considered in 1989 by Tim Berners-Lee while an analyst at CERN. Web 1.0 is a "perused as it were" web as in content suppliers and shoppers are viewed as particular gatherings and all that purchasers could do was to look for and burn-through web content given by others. There was almost no client connection with the web and with different clients and very little substance arrangement by the commonplace purchaser. Internet publicizing, e-inventories, e-pamphlets, and web based shopping baskets are all essential for the Web 1.0 experience.

As of now, an alternate phase of utilizing the web which is named as the "read-state" web by Berners-Lee is shown. Today, the recorded beneath where a significant number of us are both substance suppliers and shoppers are managed as shown below.

- Blogs – the upkeep of "web logs;" e.g., the Nudge blog
- Twitter - a "miniature contributing to a blog" administration with a limitation of 140 characters for every "tweet"; e.g., Jet Blue's utilization of tweets to answer purchaser

inquiries about flights and administration ["JetBlue Airways [JetBlue] on Twitter", n.d.]

- Mashups - sites made by clients by drawing on content from different sites, for example, raidsonline.com [a planning mashup] and bizrate.com [a shopping mashup]
- Facebook - for long range interpersonal communication; e.g., Skittles' Facebook Fan Page
- MySpace - for long range interpersonal communication yet with an accentuation on music
- LinkedIn - for proficient systems administration
- YouTube - for video sharing
- Podcasting - dispersing sound or video content to contraptions like cells, MP3 players, PCs, and work areas from web workers

This "intuitive" web is the thing that Dinucci named as Web 2.0 as far back as 1999 when such use was arising. As per Dinucci, Web 2.0 would likewise at last be recognized by its capacity to permit clients to interface with it utilizing contraptions like TVs [e.g., YouTube and Netflix access by means of AppleTV], Car Dashboard gear [for route, business repository], PDAs [for climate, route, flight notices, news], gaming consoles [for connecting players with each other over the net utilizing, e.g., Sony 's PlayStation or Microsoft's XBox], individual computerized associates [palmtop PCs or PDAs, for example, the iPod Touch], and so forth, altogether marked "compact, web-prepared" gadgets.

The equipment, interface, and execution attributes of every gadget are very not the same as the others. However, Web 2.0 would be open from these various stages separated from work area

machines running internet browsers like Firefox, Explorer, Safari, and Chrome. [24] Noticed that essentially all Web 2.0 applications are cloud applications. According to this viewpoint, Cloud Computing applications include Web 2.0 applications and Cloud Computing offers horde apparatuses that empower the easy development and conveyance of Web 2.0 applications.

Further, Web 2.0 defenders note that the term indicates not simply a set specialized details for a "new/improved" Web, however addresses a bunch of financial, social and innovation drifts that on the whole structure the reason for the cutting edge Internet. In that capacity, the continuous Cloud Computing pattern, which unmistakably is a socio-innovative pattern driven to a great extent by financial contemplations, might be viewed as an empowering influence of Web 2.0. Finally, endeavors are in progress for a push toward what Berners-Lee terms the "read-compose execute" web or Web 3.0. Web 3.0 tries to:

- i. Transform the whole web into a dispersed, worldwide information base framework lodging not just organized substance as present day information driven sites require yet in addition less organized substance like messages, archives, sound, pictures, and video;
- ii. Rely intensely on man-made consciousness [canny specialists, specifically] and regular language handling in helping clients with search in a setting touchy and customized way [the iPhone Siri Intelligent Personal Assistant is a model],;
- iii. Transform the web into a Semantic Web whereby ontological meta information to help canny specialists is inserted in the web alongside related web content information, and
- iv. Rely vigorously on ongoing, 3-D substance show utilizing the ISO X3D record configuration and XML [Extensible Markup Language] to offer rich, visual correspondence of web content where relevant.

In our view, large numbers of the continuous improvements in Cloud Computing may likewise be viewed as empowering agents of the Web 3.0 vision.

1.14. IMPLIED BENEFITS OF CLOUD COMPUTING

The current undeniable degrees of interest in, and the positive press delighted in by, Cloud Computing are energized to a limited extent by its apparent advantages. Perry [2008] takes note of that the ideal Cloud Computing foundation would have the accompanying beneficial attributes:

- Self-mending: in the event of disappointment, there will be a hot reinforcement occasion of the application prepared to take over without interruption. At the point when reinforcement turns into the essential, the framework dispatches reinforcement.
- SLA-driven: The framework is powerfully overseen by administration level arrangements that characterize strategies, for example, how rapidly reactions to demands should be conveyed.
- Multi-tenure: The framework is implicit a way that permits a few clients to impart foundation to shared darkness and without compromising protection and security.
- Service-arranged: The framework permits making applications out of discrete, re-usable, inexactly coupled administrations. Changes to, or disappointment of, an assistance won't upset different administrations.
- Virtualized: Applications are decoupled from the hidden equipment. An application might depend on Grid Computing and different applications could run on a solitary PC.
- Linearly Scalable: The framework will scale typically and productively [directly] with developing interest.

- Data, Data, Data: The way to a significant number of the above helpful attributes is the executives of information: its circulation, apportioning, security and synchronization utilizing advancements like Amazon's SimpleDB [enormous scope social data sets] and in-memory information networks.
- As per chief Cloud Computing seller, Salesforce.com, given an appropriately carried out Cloud Computing arrangement, a customer should encounter a few or the entirety of the accompanying advantages:
- Proven Web-administrations incorporation: Cloud Computing innovation is a lot simpler and faster to coordinate with other undertaking applications.
- World-class administration conveyance: Cloud Computing frameworks offer a lot more prominent versatility, complete debacle recuperation, and noteworthy uptime numbers.
- No equipment or programming to introduce: Cloud Computing requires essentially lesser capital consumptions to get fully operational.
- Faster and lower-hazard sending: No additional holding up months or a long time and burning through huge number of dollars before anybody will sign into your new arrangement. Distributed computing innovation applications are alive very quickly or months, even with broad customization or joining.
- Support for profound customizations: The Cloud Computing framework not just permits profound customization and application setup, it saves every one of those customizations in any event, during overhauls and with advancing requirements, subsequently opening up hierarchical IT assets.

- Empowered business clients: Cloud figuring innovation permits on-the-fly, point-and-snap customization and report age for business clients, so IT doesn't invest huge energy on such errands.
- Pre-fabricated, pre-incorporated applications: Hundreds of pre-constructed applications and application trade abilities are either pre-coordinated into bigger, off-the-rack applications or accessible for fast combination to shape new applications.

1.15. DIFFICULTIES FACING CLOUD COMPUTING

1.15.1 Challenges according to a Provider Perspective

Like any innovation before it, Cloud Computing, should submit to the NFLT [No Free Lunch Theorem!], its positives in any case. As indicated by a 2012 study of senior chiefs at 179 Cloud specialist co-ops by KPMG Int'l [Krigsman, 2012], there are a few difficulties that suppliers see in encouraging boundless reception/sending of Cloud Computing as Figure 2 portrays:

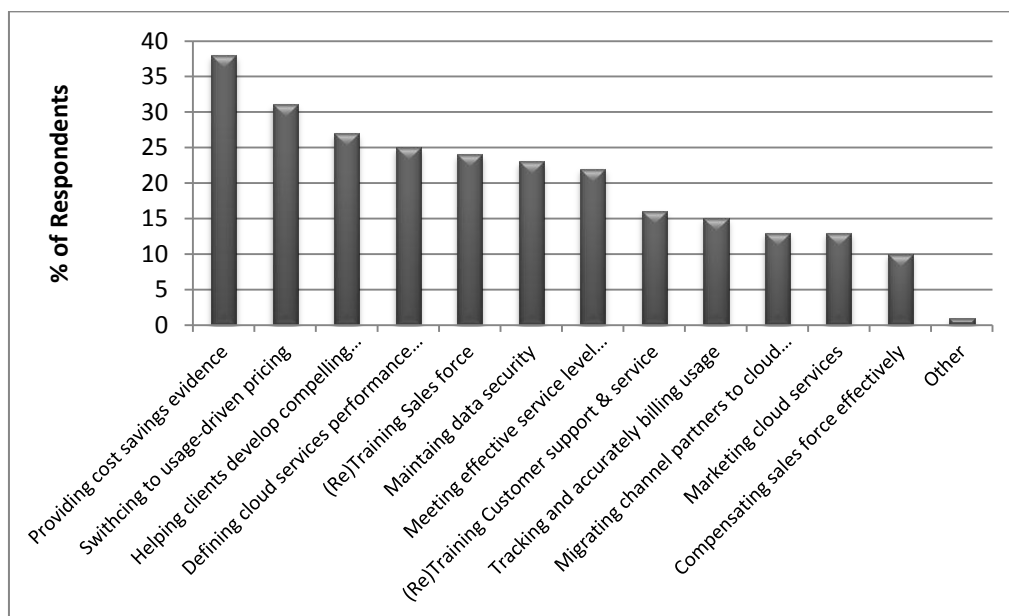


Fig.2.Shows the challenges present in cloud provisioning

See that the main three saw difficulties identify with building up an offer. The study additionally recognizes apparent disarray with respect to Cloud Services clients: "Not exactly 50% of the suppliers in the review feel their clients are educated or all around educated about Cloud Computing at the leader level. Just 43% accept clients know about cloud costs opposite their current IT administrations, and a comparable extent feels they don't completely comprehend cloud security; estimating models, mix with existing foundation and administration level arrangements [SLAs]."

1.15.2. Difficulties according to a Customer Perspective

Apparently even as associations seem energetic about Cloud Computing as a general rule, and many demonstrate that starting cloud applications is a significant need, a few appear to hold onto instabilities about considering going all in. IDC gauges that worldwide spending on Cloud Computing will be \$148 billion by 2014 and K. Bailey [CMO, Verizon Enterprise Solutions], puts the figure nearer to \$600-\$750 billion. While this addresses just about 20% of assessed spending on every last bit of IT in 2014, the projections for what's to come are considerably more bullish.

The InfoWorld Cloud Computing Special Report, 2012, recognizes nine difficulties looked by clients new to Cloud-based IT application sending [i.e., proceeded with Cloud Computing advancement]:

- Developers may find that inheritance designs utilized underway are difficult to port over to a cloud climate, without huge adjustments, for running tests against a recently created, same, Cloud-based application. [To facilitate the cycle, however, sellers like iTKO have arisen with contributions like Lisa that help with moving heritage applications to the cloud].

- High-end applications that have outrageous information security or administrative limitations, or depend on inheritance [e.g., Cobol-based] coding projects, aren't appropriate for cloud advancement.
- Cloud Computing could be a problematic innovation. In house engineers frequently loathe disturbance to set ways and incline toward working with recognizable toolsets. Top administration consolation, preparing, and, if fundamental, staff changes, are alternatives to investigate in encouraging acknowledgment.
- There is a shortage of documentation to assist engineers with understanding Cloud-based instruments and their utilization. This could change with expanding reception or associations should enlist outside advisors for help.
- Unless an association is cautious, its designers could undoubtedly neglect to wind down Cloud-based administrations when not required and superfluously run up rental charges.
- Organizations should be clear about authorizing concurrences with cloud sellers or be disagreeably astounded with what they can or can't achieve with cloud-based assets.
- Cloud engineers generally don't have open admittance to the supplier's framework, applications, and mix stages. This could present difficulties in incorporating in-house applications with cloud-based applications and surprisingly in coordinating different cloud-based applications. Dependence on suppliers who make accessible proper Application Program Interfaces [APIs] is significant.
- Cloud Computing is in a transformative state and the speed of progress is fast. Associations should work with sellers to stay informed concerning best practices as they quickly develop.

1.15.3 A Note on Security, Interoperability, and Portability Concerns

Concerns identifying with security, interoperability, and versatility in cloud-based assistance arrangements invade both specialist co-ops and shoppers and the NIST takes note of that these worries are the best three boundaries to more extensive cloud administrations reception. From the purchaser side, giving up information and applications control to an outsider, offering IT assets to different customers, and the lock-in relationship with a cloud supplier are significant contemplations that they should weigh when adjusting the danger and advantages of embracing Cloud Computing. To meet general IT security prerequisites, associations need to have security control arrangements to ensure and safeguard the respectability, accessibility, and classification of IT assets from unapproved access, disturbance of administration, burglary, abuse, malignant assaults, and such. Distributed computing presents exceptional security challenges that go past these overall security prerequisites.

These additional difficulties stem basically from two sources: the utilization of Virtualization and a multi-inhabitant climate inborn to Public, Community, and Hybrid cloud conditions [see Section 5 for depictions]. Virtualization assists with cloud adaptability. In any case, Virtualization makes added information trustworthiness and privacy issues because of weaknesses in hypervisors. A compromised hypervisor might actually harm all frameworks that it has. Another virtualization related security challenge is to isolate the information, areas, virtual machines, API calls, exchanges, and so on, of each inhabitant in a multi-occupant setting from those of different occupants for secrecy and uprightness affirmation.

An interloper can access not just one customer's information/applications on the cloud however every other customer's information/applications also by basically taking advantage of blemishes in the cloud's multi-occupancy plan or uncertain APIs. In fact, information breaks are considered the most genuine of safety dangers with Cloud Computing [Cloud Security Alliance, 2013]. A supplier should forestall a solitary security break from affecting the whole cloud climate. The scale and intricacy of the organization design basic Cloud Computing further confounds

endeavors to comprehend security weaknesses to appropriately address and diminish dangers to worthy levels. Distributed computing, with its many interconnected parts of equipment, programming, information, and media communications, must be completely interoperable to convey a consistent progression of data and administrations. To be completely interoperable clients need to can incorporate and solidify their information, applications, and frameworks across cloud foundations and among cloud suppliers.

The thought for an "Intercloud," where Cloud Computing administrations including information, stockpiling, processing, and so on, are pervasive and interoperable in a Web-based organization of mists across various areas, stages and topographies, is at present being examined under a joint exploration project among IBM and the European Union called "Repository" or Resources and Services Virtualization Without Barriers [Sacharen and Hunter, 2008]. New Virtualization and Grid advancements will be expected to empower the degree of interoperability imagined in Intercloud. Other mechanical difficulties that should be addressed incorporate the capacity to oversee and convey huge scope responsibilities to advance financial and nature of administration prerequisites arrange and indicate administration level arrangements, just as plan and deal with an organization of more than 100,000 server farms crossing enormous geographic distances.

Transportability is the adaptability to move information and applications starting with one cloud supplier then onto the next or among private and public cloud conditions [www.webopedia.com]. This adds up to loosening up the lock-in necessity with a particular cloud specialist organization so clients have full oversight of their information or potentially applications. The capacity to import/trade huge volumes of information, the possession the board and access control of shared or cooperative information, just as the security and validation instruments required to help this degree of convey ability stay an innovative test of Cloud Computing.

1.16. FINISHING UP REMARKS

In this part, the historical backdrop of Cloud Computing from its initial roots were followed in Time Sharing with centralized computer PC frameworks, through Peer-to-Peer and Client-

Server figuring, Grid Computing, Utility Computing, Virtualization, and Service-situated Architecture. Then, at that point presented Cloud Computing as a particular advance in this development where the accentuation is on simplicity of arrangement of different administrations to clients. Following this conversation, four models of Cloud Computing organizations were discussed in particular, Private, Public, Hybrid, and Community mists. At that point, our concentration toward a wide range of sorts of administrations given by cloud specialist organizations and summed up key parts of the various administrations accessible today are directed.

Given our comprehension of cloud-based administrations, Cloud Computing was clarified which plays as an empowering agent of the present Web 2.0 and the future Web 3.0. The assessment of Cloud Computing was balanced by articulating the advantages accruable and the difficulties one appearance with Cloud Computing. Notwithstanding its extensive developmental history, the field of present-day Cloud Computing is at this point in its early stages and is going through a lot of early stage problems. At the hour of this composition, two significant cloud administration blackouts were standing out as truly newsworthy. Netflix, the online video specialist organization experienced assistance blackout in the US East Region, through a backend-disappointment Christmas eve through Christmas day, 2012 [24]. Netflix was facilitated on Amazon's Backend-as-a-Service [Baas] cloud.

To confound matters, Netflix has been let down threefold, until now, by its cloud administrations supplier, Amazon [21]. Already, a July 2012 Amazon blackout affected Netflix, Pinterest, and Instagram. The next week, on Dec 28, 2012, Microsoft's Azure cloud-based capacity administration for the South-Central US encountered fractional blackout. The STaaS blackout, at first expected to be settled in a couple of hours, proceeded more than 50 hours. Solutio, an organization that runs a PC diagnostics apparatus for Windows clients around the world, had relocated to Azure in 2010 after powerlessness of its private stockpiling framework to deal with unexpected burden spikes.

These are a couple of ongoing models. A large number of us have encountered cloud-based email administration [EaaS] blackouts with suppliers like Hotmail, Yahoo, and Gmail throughout the long term. While the standpoint for relocation to cloud administrations looks bullish as per the cloud savants, episodes, for example, these should make planned customers stop, re-consider, and tread carefully. In any event, strategic applications may not be prepared for the cloud without broad and costly safeguard gauges set up. Also, customers should acknowledge conceivable cloud-administrations disappointments as a basic piece of running business on the cloud, much as they should acknowledge unavoidable administrations blackouts for their non-cloud-sent administrations.

REFERENCES

1. Anshari M, bin Alas Y, Guan LS. Pervasive knowledge, social networks, and cloud computing: e-learning 2.0. *EURASIA Journal of Mathematics, Science and Technology Education*. 2017 Jun 16;11(5):909-21.
2. Ari AA, Ngangmo OK, Titouna C, Thiare O, Mohamadou A, Gueroui AM. Enabling privacy and security in Cloud of Things: Architecture, applications, security & privacy challenges. *Applied Computing and Informatics*. 2020 Jul 31.
3. Anand A, Chaudhary A, Arvindhan M. The Need for Virtualization: When and Why Virtualization Took Over Physical Servers. In *Advances in Communication and Computational Technology 2021* (pp. 1351-1359). Springer, Singapore.
4. Abubakar A, Babate AI, Ishola A. The Study of Data Security in Cloud Computing. *European Journal of Electrical Engineering and Computer Science*. 2020 Aug 31;4(4).
5. Ayankoya FY, Agbaje MO, Ohwo BO. Appraisal on Cloud Computing and Network Functions Virtualization. *IJCSNS*. 2019 Jul;19(7):38.
6. Bokhari MU, Makki Q, Tamandani YK. A survey on cloud computing. In *Big Data Analytics 2018* (pp. 149-164). Springer, Singapore.
7. Brinda M, Heric M. The changing faces of the cloud. Bain Company. 2017.
8. Christov Y. Cleaning Data with Cloud Based Platform Toolsets. In *Proceedings of International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE) 2017* (pp. 503-509). International

Conference on Application of Information and Communication Technology and Statistics and Economy and Education (ICAICTSEE).

9. Chen J, Zhu Q. Security as a service for cloud-enabled internet of controlled things under advanced persistent threats: a contract design approach. *IEEE Transactions on Information Forensics and Security*. 2017 Jun 21;12(11):2736-50.
10. Dinakar KR. A survey on virtualization and attacks on virtual machine monitor (VMM). *Int. Res. J. Eng. Techn.*. 2019 Mar;6(3):6558-63.
11. Easttom C. Computer security fundamentals. Pearson IT Certification; 2019 Oct 2.
12. Enberg A, Foleti O. Creation of a private cloud infrastructure: building a foundation for cloud services.
13. Foster I, Gannon DB. Cloud computing for science and engineering. MIT Press; 2017 Sep 29.
14. Hurwitz JS, Kirsch D. Cloud computing for dummies. John Wiley & Sons; 2020 Jul 7.
15. Jain A, Mahajan N. Introduction to Database as a Service. In *The Cloud DBA-Oracle* 2017 (pp. 11-22). Apress, Berkeley, CA.
16. Jain SM. Virtualization Basics. In *Linux Containers and Virtualization 2020* (pp. 1-14). Apress, Berkeley, CA.
17. Kasemsap K. Software as a service, Semantic Web, and big data: Theories and applications. In *Research Anthology on Recent Trends, Tools, and Implications of Computer Programming 2021* (pp. 1179-1201). IGI Global.
18. Kaur A, Gupta P, Singh M, Nayyar A. Data placement in era of cloud computing: a survey, taxonomy and open research issues. *Scalable Computing: Practice and Experience*. 2019 May 2;20(2):377-98.
19. Krigsman M. Cloud research: Cost matters most and confusion remains. Retrieved on. 2013;23(9).
20. Le Dinh T, Dam NA. Smart Data as a Service. In *ITM Web of Conferences 2021* (Vol. 38, p. 03001). EDP Sciences.
21. Nider J. A comparison of virtualization technologies for use in cloud data centers. IBM research report H-0330 (HAI1801-001); 2018.

22. Niknejad N, Amiri IS. Literature review of service-oriented architecture (SOA) adoption researches and the related significant factors. The impact of service oriented architecture adoption on organizations. 2019:9-41.
23. Oppitz M, Tomsu P. Cloud computing. In *Inventing the Cloud Century* 2018 (pp. 267-318). Springer, Cham.
24. Petrenko S. Big Data Technologies for Monitoring of Computer Security: A Case Study of the Russian Federation. Springer International Publishing; 2018 May 17.
25. Pujari MV, Sharma Y, Jangam MS. NEED OF VIRTUALIZATION.
26. Rashid A, Chaturvedi A. Virtualization and its role in Cloud Computing environment. *International Journal of Computer Sciences and Engineering*. 2019 Apr;7(4):1131-6.
27. Rajeswari S, Kalaiselvi R. Survey of data and storage security in cloud computing. In *2017 IEEE International Conference on Circuits and Systems (ICCS)* 2017 Dec 20 (pp. 76-81). IEEE.
28. Sharma M, Husain S. Analyzing the Difference of Cluster, Grid, Utility & Cloud Computing. *IOSR Journal of Computer Engineering*. 2017;19(1):55-60.
29. Smith J, Smith C. *Adobe Creative Cloud All-in-one for Dummies*. John Wiley & Sons; 2017 Dec 4.
30. Tayade V, Khan MR. A STUDY ON CLASSIFICATIONS OF CLOUD. *International Journal of Engineering Technologies and Management Research*. 2019 Dec 31;6(12):68-72.
31. Wang T, Zhang G, Liu A, Bhuiyan MZ, Jin Q. A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing. *IEEE Internet of Things Journal*. 2018 Sep 13;6(3):4831-43.
32. Yi B, Wang X, Li K, Huang M. A comprehensive survey of network function virtualization. *Computer Networks*. 2018 Mar 14;133:212-62.
33. Zhou Y, Zhang D, Xiong N. Post-cloud computing paradigms: a survey and comparison. *Tsinghua Science and Technology*. 2017 Dec 14;22(6):714-32.

CHAPTER-2

VIRULIZATION IN CLOUD COMPUTING: EXISTING SOLUTION AND NEW APPROACHES

2.1 INTRODUCTION

When virtualization was first studied in the 1970s, it was known to as time sharing by scientists and software engineers. Multiprogramming and relative thinking became the driving forces behind the development of a few PCs, including the Atlas and IBM's M44/44X. The Mapbook PC was one of the first supercomputers to use concepts like time sharing, multiprogramming, and shared fringe management in the mid-1960s. The ChartBook is the fastest PCs due to a partition of the operating system from the running client programmes. The director segment associated with the PC's time management and passed extra codes along these lines, assisting with the client program's rules administration. The hypervisor, or virtual server screen, was thought to have been introduced at this time [1].

2.1.1 Virtualization

Where the framework separates the source into one or more implementation contexts, virtualization [2] refers to the construction of a virtual copy of a gadget or source, like a webserver, storage source, internet, or even an user interface. To put it another way, virtualization is a structure or methods for splitting a computer's resources into numerous hardware platforms using one or more concepts or innovations like software and hardware partitioning, time-sharing, incomplete or complete machine visualisation, service quality, amplification, etc (Figure 2.1).

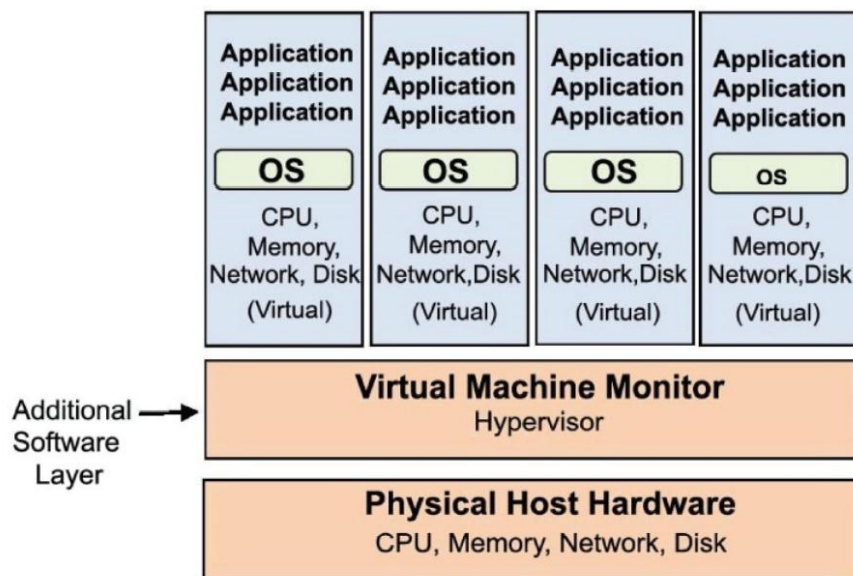


Figure 2.1 virtualized system

2.2 IMPORTANCE OF VIRTUALIZATION

There are numerous advantages of LM. It allows the client the flexibility and options to shut down a running server in the middle of the day, rather than late at night or on weekends, to revamp the working structure, apply fixes, and so on; it can then be replicated during normal working hours. This is a fantastic idea; for instance, operations management in server farms look at where they have large workloads and shift VMs around so the cooling framework doesn't have to work as difficult to keep a portion of the data centre at the proper temperature. There are two basic aspects to virtualization: Process VMs and System VMs are two types of virtual machines (Figure 2.2)

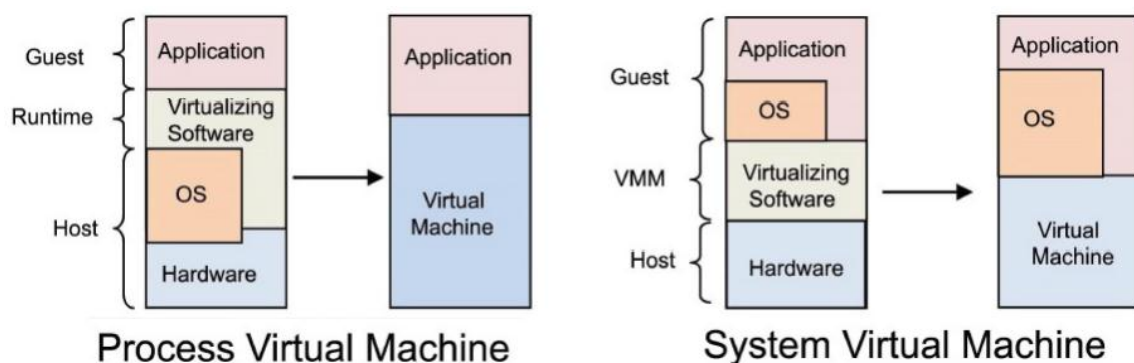


Figure 2.2 virtual machines types

2.3 BENEFITS OF VIRTUALIZATION

1. Virtual machines are used to consolidate the workloads of a large number of underutilised servers onto a single system.
2. Other beneficial situations include equipment reserve funds, natural costs, server administration, and server foundation organisation.
3. Virtual machines are ideal for running legacy programmes.
4. Virtual machines (VMs) can be used to provide safe, separate sandboxes for running non-stock-in applications. Virtualization is an important concept in creating safe figuring stages.
5. VMs are used to create functional frameworks or implement conditions with asset ports of limitation, and they can also be used to ensure assets if the correct schedulers are used.
6. VMs can provide you with a piece of machinery or a set of equipment that you don't have example, SCSI gadgets & different processors.
7. Virtual machines (VMs) are used to operate many agent systems at the same time: different forms, or altogether different frameworks, can be on hot standby.
8. Virtual machines provide for efficient research and execution observation.
9. VMs can detach what they're running in order to provide responsibility and error control. Virtual machines (VMs) make it easier to transfer programming around, allowing for greater application and framework flexibility.
10. Virtual reality headsets are fantastic tools for exams and academic challenges.
11. Virtualization can be used to modify current agent frameworks so that they can continue to run on shared memory multiprocessors.
12. Virtual machines are used to create self-assertive check circumstances, which can lead to some incredibly innovative and effective quality assurance.

13. Virtualization can make tasks like framework migration, reinforcement, and recovery less difficult and more cost-effective.

14. Virtualization is a good way to achieve twofold similarity.

2.4 TYPES OF VIRTUAL MACHINES

Process virtual machines and system virtual machines are the two types of virtual machines (Table 2.1).

Table 2.1 virtual machines types

Process Virtual Machine	System Virtual Machine
Virtualizing software translates instructions from one platform to another platform	It provides a complete system environment
It helps execute programs developed for a difference operating system or difference ISA	Operating system + User Process
Virtual machine terminates when guest process terminates	Networking + I/O + Display + GUI
	Lasts as long as host is alive

2.5 VIRTUAL MACHINE APPLICATIONS

Figure 2.3 and Table 2.2 show virtual machine applications.

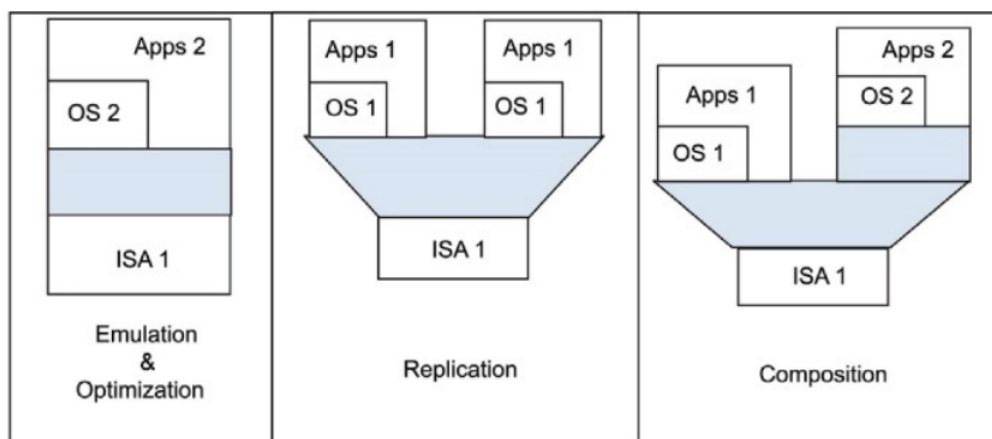


Figure 2.3 Applications for virtual machines

Table 2.2 Applications for virtual machines

Applications	Description
<i>Emulation</i>	It permits blend and match cross-stage convenience.
<i>Optimization</i>	It gives stage particular execution change. It is normally finished with imitating.
<i>Replication</i>	This permits having various VMs on a solitary stage.
<i>Composition</i>	Similar to replication yet shapes more mind-boggling yet adaptable frameworks.

2.6. VIRTUALIZATION COMPONENTS

The hypervisor [2], or virtual machine administrator, and the guest are two fundamental components of virtualization. The hypervisor is in charge of the virtualization layer's management. Type-1 and Type-2 hypervisors exist. Figure 2.4 illustrates Type-1 and Type-2 hypervisors.

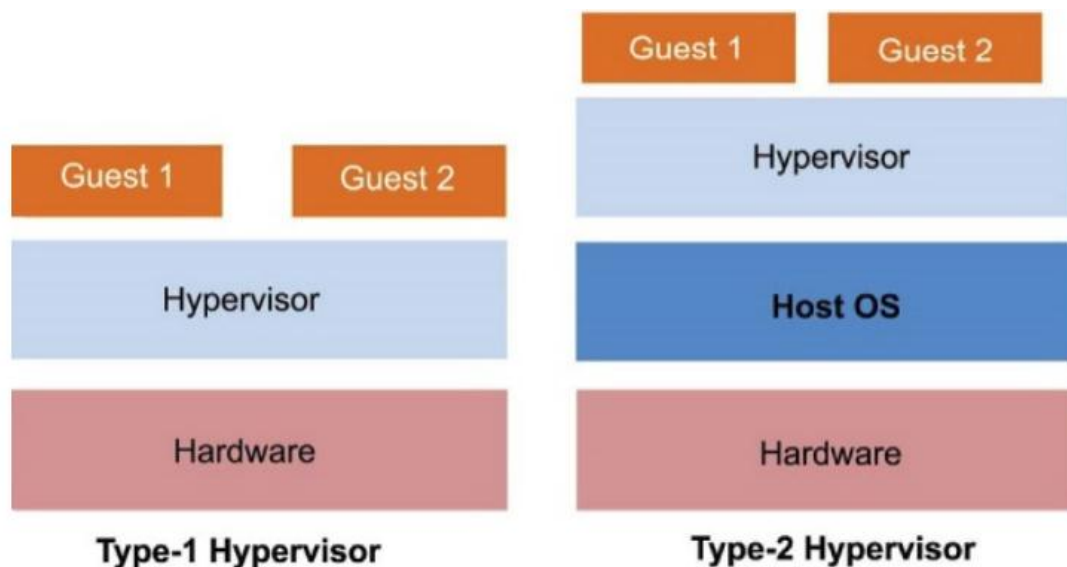


Figure 2.4 Type-1 and type-2 hypervisor Type-1 hypervisor,

It has immediate access to the computer's hardware. Because of its design, it offers more flexibility and higher performance. Type 1 hypervisors include Xen, Microsoft Hyper-V, and VMware ESX. As a program, a Type-2 hypervisor runs on top of the host machine's OS. There was one virtualization screen to operate and manage each virtual machine, hence there is only one virtual environment for each virtual environment to work on. Type-2 hypervisors include

VirtualBox, VMware Player, and VMware Workstation. The digital host that operates atop the virtualization layer is known as the guest. The virtual host runs on its own OS and software.

2.7. VIRTUALIZATION TYPES

VMware is primarily portrayed as "the partition of an admin asks for from the hidden actual delivery of that admin," that's one of the world's largest businesses now accumulating practical expertise in virtualization. What VMware means by "hidden delivery" of an organization is that an intermediate takes care of the precise implementation of the guideline begun by a VM, such as a processor request or a memory transaction. Depending on the method of virtualization used as part of the system, the VM might be aware that this is happening. A virtualization phase can be created in a variety of ways. There are a few commonly used terminology that need to be clarified in order to understand the distinctions and subtleties underlying each of them: A virtual machine that runs on top of a virtualized environment is referred to as a guest. The guest is usually a completely functional operating system in and of itself. The host is a computer that allocates resources to a visitor. Between the physical server and the host, the hypervisor serves as an abstraction layer. A hypervisor was sometimes known as a "virtual machine manager" (VMM). This is the proxy who acts on behalf of the guest to carry out orders. Full virtualization, paravirtualization, and hardware-assisted virtualization are the three types of virtualization.

2.7.1 Full Virtualization

This is the most straightforward and straightforward sort of virtualization. A hypervisor is simply installed on a physical device that has at least one of each of the usual PC components: a network card, a hard disc, a processor, and memory (RAM). The type of hypervisor used here is an exposed metal hypervisor, often known as "sort 1." The visitor's operating system is unmodified, and hence perceives the devices as "real," implying that the OS is unable to check that it is being simulated. Off camera, the PM assigns assets to the guest OS. This type of operation could be more adaptive and productive if the hypervisor had guided access to the devices. In any event, this virtualization mode relies on the visitor and host interpreting CPU rules in simultaneously, as it reenacts all secret equipment to the guests, which can have a significant impact on productivity in terms of overhead. Visitors can see the entire design of a

PC framework thanks to full virtualization. If security is a major concern, it is recommended that you use a full virtualization suite, which can provide cautious disconnection of operating apps. The hypervisor can split assets into groups, and it's common to only activate one pool per VM and only host one application per VM. A security breach event can do a lot of damage if a PM is operating a lot of VMs that support a lot of different apps.

2.7.2 Para Virtualization

Similarly, configuring virtualization necessitates altering the visitor's Software and including the host's hypervisor layer. At that moment, the hypervisor sits between the visitors and the host. This necessitates a "Type 2" hypervisor, also known as a facilitated hypervisor, and thus this type of hybridization is referred to as facilitated hypervisor. It's essentially software that runs in the background on the host operating system. Because changes to the visiting OS are required, each VM might be aware that it is being simulated. With today's OS pictures, this happens naturally during setup. The visitor's asset queries require them to interact with the hypervisor on the host before they can access the tangible assets, which might add to the overhead. Nonetheless, because the VMM is small and simple to paravirtualize, visitors can usually achieve "close local" performance. This means that virtual machines are close to executing instructions as quickly as a physical computer with equivalent specifications. Visitors do not access assets through duplicated devices, as in complete virtualization, but rather via special device drivers [5, 6]. Advantage rings are a concept in the x 86 designs (Figure 2.5). The executions of an operating system must take place in ring-0. When a VM initiates such a procedure (known as a hyper call), the virtualization running alongside the host OS detects it and executes it for the visitor's advantage [7].

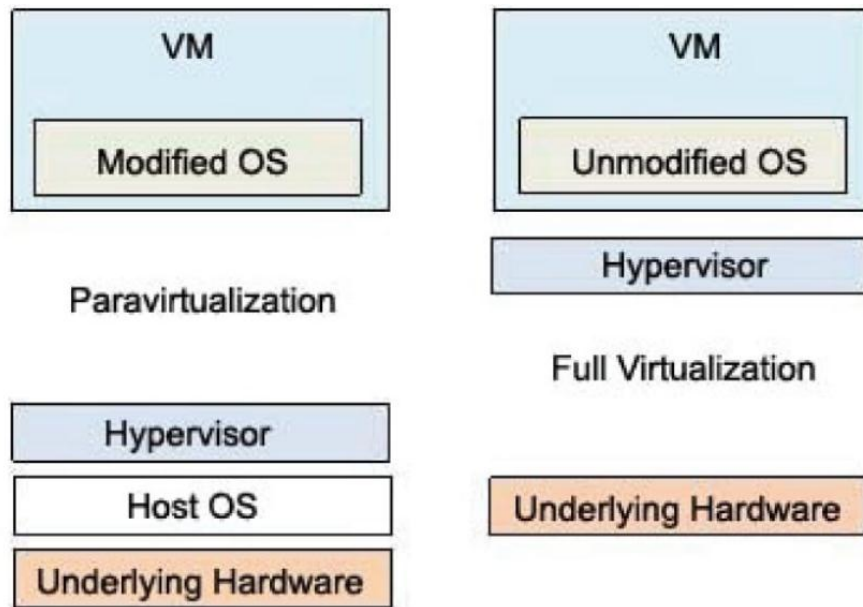


Figure 2.5 design of full- and para virtualization

2.7.3 Hardware-Assisted Virtualization

Enhanced virtualization is another name for hardware-assisted hypervisor. Using the particular features afforded by computer hardware, users can use an unmodified processor. Since 2006, both AMD and Intel have supported hardware virtualization. The virtual machine device is operated on a root mode pyramid level below ring-0 in this approach. In other words, hardware virtualization adds a new privilege level below ring-0 for the hypervisor, leaving ring-0 for the unmodified guest os. Figure 2.6 depicts the defensive ring topology of hardware-assisted emulation. This system wasn't compatible with CPUs built prior to 2006.

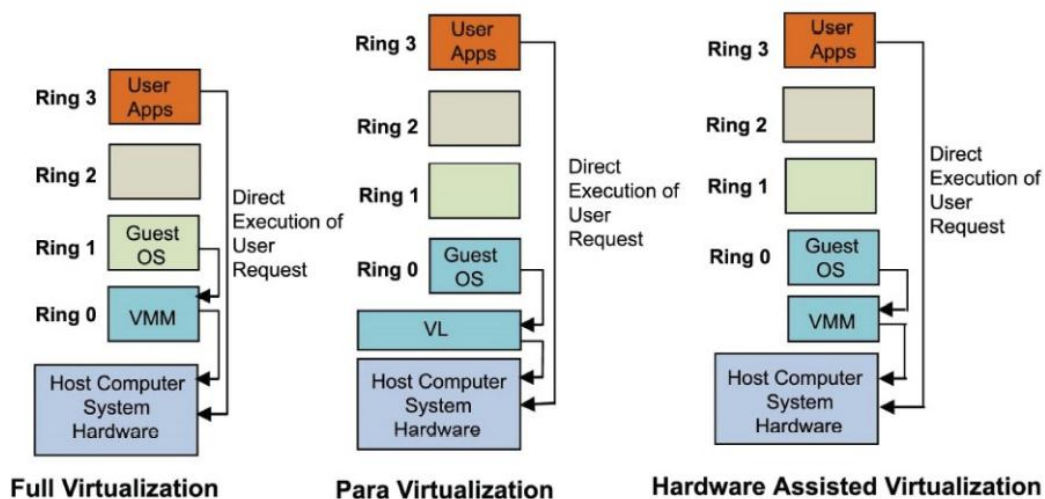


Figure 2.6 virtualization types.

2.8 VIRTUALIZATION SYSTEM

2.8.1 Xen Hypervisor

Xen, as seen in Fig.2.7, is a web based hypervisor that supports both hardware-assisted hypervisor and para-virtualization and was developed at the University of Cambridge computer science lab. It also enables the LM of virtual machines. Xen was one of the web services VM1 type-1 or bare-metal hypervisors available. Multiple operating systems can operate in parallel with the host operating system thanks to Xen. Server virtualization, desktop virtualization, IaaS, privacy, and hardware and integrated appliances are just a few of the open source and proprietary applications that employ Xen. Today, the Xen hypervisor is used to power massive commercial clouds. The Xen hypervisor is in charge of interrupt management, CPU planning, and memory management for virtual machines.

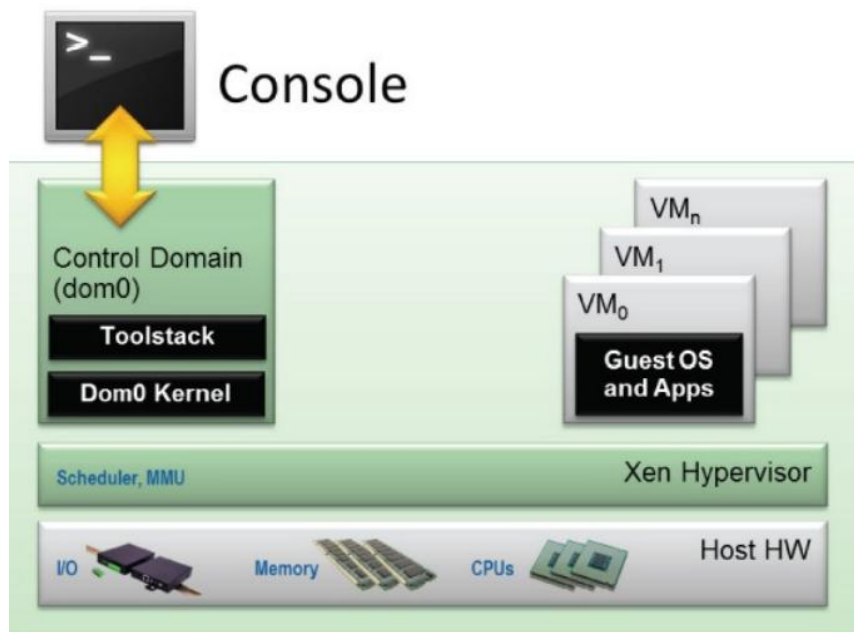


Figure 2.7 architecture of Xen .

Dom0, or Domain-0, is the zone in which Xen begins its boot process. Dom0 is the authorized control domain with direct access to the physical infrastructure, as seen in the Xen design shown in Figure 2.7. The toolstack, which is a user control interface to the Xen hypervisor, is installed

on Dom0. The Xen toolstack can construct, manage, and remove virtual machines (VMs), sometimes known as unprivileged domains (domUs). Hardware virtualization and paravirtualization are supported by Xen. Hardware virtualization allows for the use of unmodified operating systems for virtual servers, while paravirtualization necessitates changes to the kernel of the linux kernel running inside operating systems. The performance of paravirtualized hosts will improve as a result of this. To construct VMs, the host operating system must support Xen PV (paravirtualization) [3]. Paravirtualization is not supported in Linux kernels prior to version 2.6.37. To allow paravirtualization, their processors must be recompiled. Xen PV is installed by default in all Linux kernels published after version 2.6.37 [4].

Phy and file are the two advantages of electronic block devices that Xen offers. Phy refers to the physical block device that exists in the host context, whereas file refers to the disc image that exists on the host machine as a file. The loop block device is created using the supplied file type, and the domU manages the block device. Phy is used by shared storage systems like iSCSI, and file is used by NFS [5].

2.8.2 KVM Hypervisor

KVM's design is made up of two parts: a kernel and a userspace component (Figure 2.8). The KVM kernel unit is the kernel module that exposes the virtualization features to the data plane via a character device called `/dev/kvm`. This device node executes the guest program on the host equipment. For the implementation of guest program, KVM introduces a new guest style of execution to the kernel as well as user capabilities on the host kernel. KVM offers the essential virtualization architecture in this fashion, but it does n't offer any device simulation or VM administration. This is managed via the QEMU-KVM backend, a slightly altered QEMU (Quick EMUlator) process, which is a user-space element. It functions similarly to standard QEMU in terms of device simulation, generating VMs, relocation of VMs, etc; but, instead of totally simulating vCPUs, it executes them as normal Linux strands in the host's data plane and employs the guest activation mode. Each guest's vCPU is implemented as a Linux process. All hosts in this architecture appear to be regular Linux programs, and the host kernel scheduler manages the allocation of these vCPU units. One benefit of implementing guests as threads on

the host kernel was that they may now be governed from the host in terms of planning, prioritisation, and affinity.

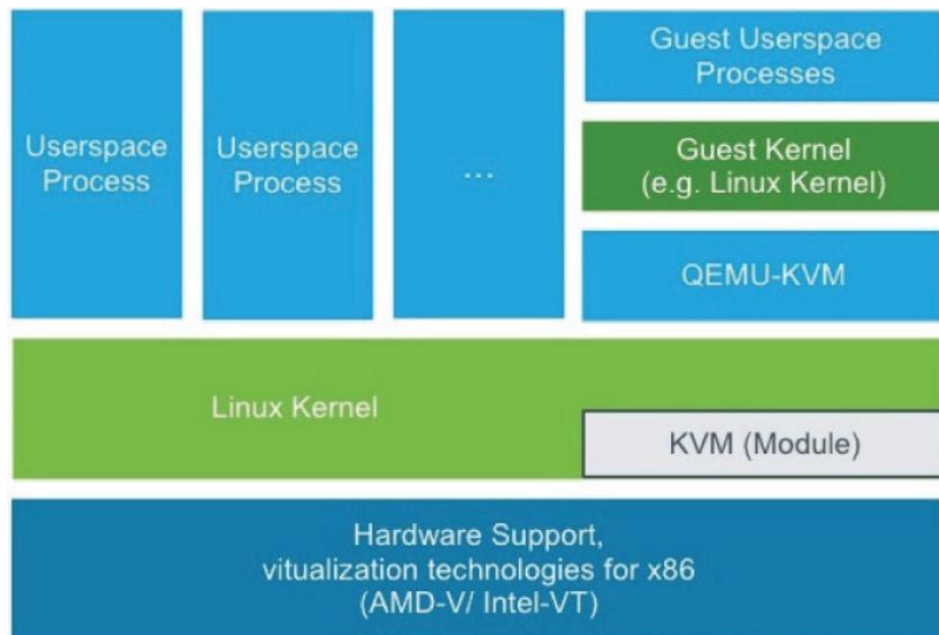


Figure 2.8 KVM Architecture.

2.8.3 OpenStack

OpenStack Kilo is the newest (11th) edition of OpenStack, a Python-based open-source cloud platform that began as a collaborative initiative between Rackspace Hosting and NASA in 2010. (Figure 2.9). It's a cloud operating system that manages huge pools of computing, network services, and storage across a data centre using a portal that offers administrators control while allowing users to provide resources via a web browser. As previously stated, OpenStack groups all of a data center's capabilities into pools of compute, space, and web servers. A web-based portal, command-line input capabilities, or RESTful API can all be used to administer these resource pools. OpenStack is easy to set up, enormously scalable, and scales to suit the needs of any cloud computing. The OpenStack design is a modular design, which means it is made up of several different components that function together. Furthermore, components in such architectures can be swapped out, deleted, or incorporated without affecting the remainder of the system.

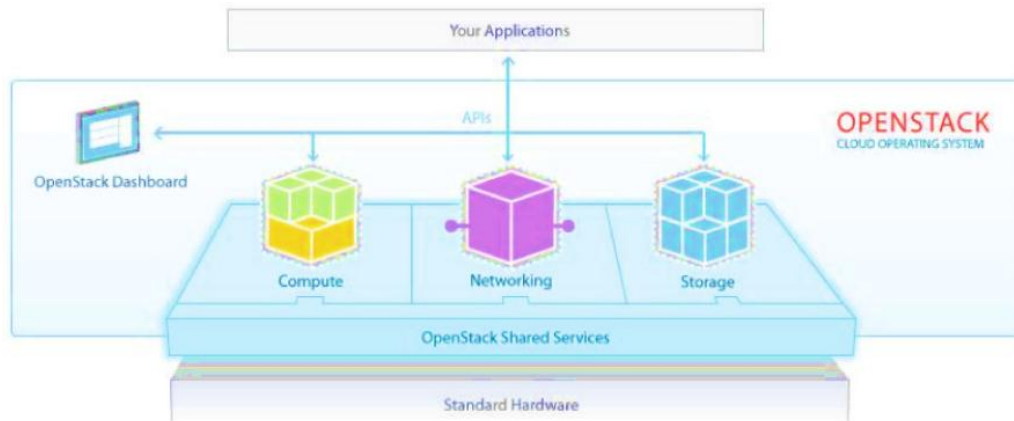


Figure 2.9 architecture of OpenStack

2.8.4 Storage

In OpenStack, there are two types of storage for instances: ephemeral storage and persistent storage. In an OpenStack setting, ephemeral memory is handled as a file on the physical server and is connected with the instance. This storage exists for the duration of the example and is removed when it is destroyed. Persistent memory lasts longer than the instance and is accessible independent of its state. Block memory (Cinder) and encrypting data (OpenStack) are the 2 kinds of storing data provided by OpenStack (Swift). Cinder, a block cloud hosting, offers running instances with permanent block storage. These configurable block devices, also known as volumes, can be configured, split, and mounted much like regular discs or partitions. Swift is a cloud-based object storage service with a multi-tenant architecture. It has a RESTful, HTTP-based API that can manage enormous volumes of unstructured data objects and is extremely accessible and scalable. It employs a distributed design that eliminates the need for a central control point, resulting in increased scalability, resilience, and stability. For data redundancy and consistency across the network, the items are copied to several hardware devices. The Glance picture service and the Cinder service can both use Swift system can store VM images and backup VM volumes.

2.8.5 Server Virtualization

Server virtualization allows several virtual machines to be created on a single physical computer. The real machine is known as the host, and the virtual machines that run on it are known as guests. The phrases virtual machines (VMs) and guests are interchangeable. A software called

the inverter, which operates on the host, assigns each VM its portion of the host's physical resources (CPU, storage, disc, I/O, and so on).

2.9 LIVE VIRTUAL MACHINE MIGRATION

In its most basic form, live virtual machine movement is the transition of virtual machines (VMs) from one physical server to another with little or no service interruption (Figure 2.10).

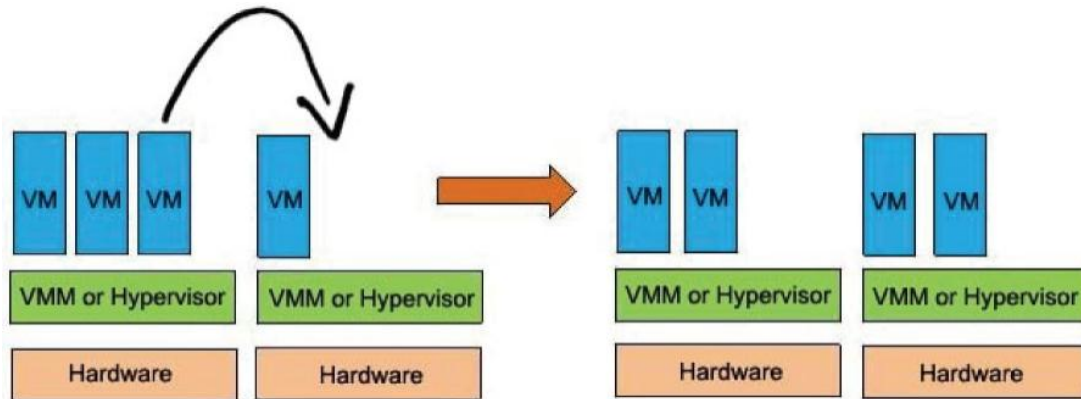


Figure 2.10 Migration of virtual machines.

Migration Forms: As stated in the last chapter, there are two main types of migration: post-copy and pre-copy. Some unique migration procedures are as follows: Migration Stop-and-Copy: Non-live migration is the term for this form of migration [6]. There will be VM downtime as an outcome of this type of transfer. It works nicely for VMs that are being maintained. The only benefit of this form of migration is that it establishes a baseline against which the total number of pages transmitted and the overall migration duration can be compared.

The below are the specifics of the stop-and-copy migration:

1. The source VM is turned off (shut down).
2. Every page is duplicated over the network.
3. Last but not least, the target VM is started.
4. This form of migration has the greatest service outage.
5. This migration method takes the smallest amount of time to complete.

Dynamic Migration: This sort of migration is focused on live VM migration using post-copy techniques. The below are the specifics of demand migration:

1. All crucial and necessary OS structures are transferred across the network at first.

The destination VM is then launched.

3. Every page failure causes a network duplicate of those pages.
4. This form of migration has the least service outage.
5. This migration type takes the longest to complete.

Iterative Pre-Copy Migration: This style of migration blends a bounded incremental push stage with pre-copy integration. This form of migration also includes a brief stop-and-copy phase. The details of iterative pre-copy transfer are as follows:

1. Iteratively transfers webpages from origin to destination over the system.
2. Continues copying webpages until a certain threshold is achieved, at which point the source VM is stopped and all remaining pages are copied before the destination VM is started.
3. This type of migration strikes a balance between processing time and service interruption..

2.9.1 QEMU and KVM

QEMU is a piece of software that allows you to emulate and virtualize machines. The application has several modes of operation, including "full system simulation," which allows it to host a wide range of computer kinds as guests. This is accomplished through a recompiler in the QEMU programme, which converts binary code intended for one CPU kind to binary code intended for another. Many more computer elements, like network cards, hard discs, and USB, are also emulated in QEMU. In other terms, QEMU is a hypervisor that provides virtualization features through emulation. KVM, which means for kernel vm, is a hypervisor that is a derivative of the QEMU programme. It's part of the Linux operating system, and when it's turned on, it turns the ordinary Linux kernel into a virtualization. It treats visitors as if they were processes on the host, allowing them to be managed just like any other application. Process IDs

are given to them, and KVM can interact with them. The KVM programme can be started from the Linux command prompt, giving the impression that it is a Type-2 hypervisor that runs “on top” of an operating system. However, KVM's virtual machines run on metal surfaces, thereby making it a Type-1 inverter. The differences between THE answers are found in which designs might be perplexing at times.

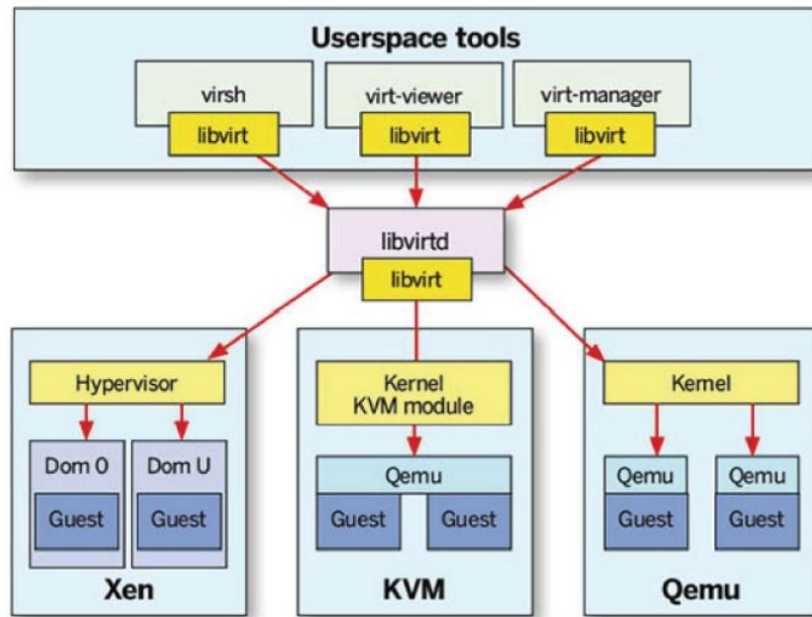


Figure 2.11 QEMU and KVM.

2.9.2 Libvirt

Libvirt was a virtualization management API and daemon created by Red Hat (see Figure 2.12). You can use the **virbr** command interface to access to VMs remotely over the network and to start VMs. Libvirt can create a virtual network gateway on the host, which guests can join to and use to send their network traffic. The virtualization layer was a feature that may be turned on or off.

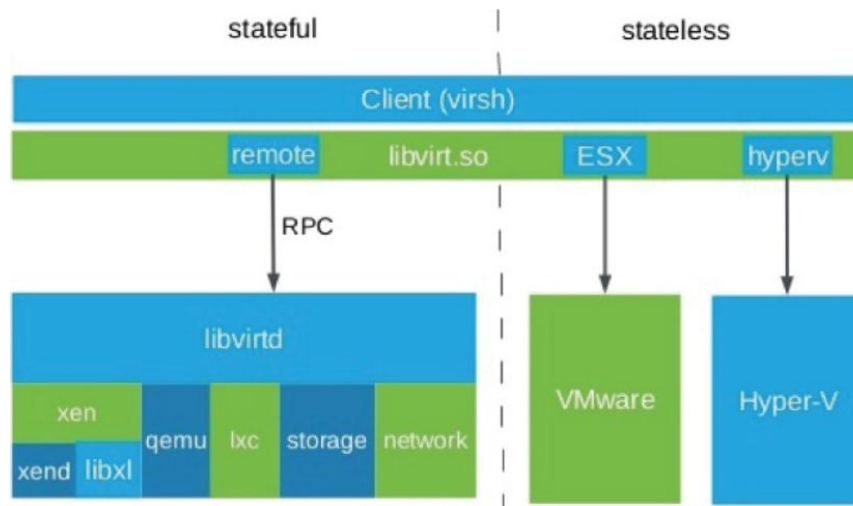


Figure 2.12 Architecture of Libvirt

By default, it implements NAT using the masquerade settings, making all VMs that link to exterior sources appear to be traffic originating from the virtual bridge's IP address . The switch programme can also run in "routed mode," which places VMs on a subnet that is hidden from the host. In this configuration, computers from the outside can connect to the VMs via a dynamic route on the server, which directs traffic to the bridge connection. A third Libvirt approach is to connect a real surface to an existent virtual circuit on the host. The existing connection would then be linked to each VM via a tap interface. This was a virtual interface that the host sees as a physical connection and the swap software sees as a switch port. It's typical to utilise a mixture of Libvirt, QEMU, and KVM to set up distributed emulation on a Linux system.

2.10 CONCLUSION

This chapter taught how to use one of the most extensively used VMMs. VMware's live migration feature is open to practical assaults. These hazards are alarming, and they necessitate the application of appropriate remedies to each type of live migration hazard. Also, by default, vMotion is done in cleartext, and VMware cryptography for vMotion isn't foolproof, as the tests show; administrators should test vMotion before putting production systems online. Admin can also attempt incorporating tried and true encryption technologies to ensure that vMotion is safe. A strong code of conduct must be followed when managing virtualization layer. It's time to rethink security measures. Any modifications to the virtualized systems must be closely

monitored, as it is easy to make changes but difficult to maintain them. Any exploit, such as a hijacked guest VM, can be utilised elsewhere, and if it's a database system, for example, nothing can prevent the thief from taking data. Furthermore, by focusing on availability, it is possible to limit VM interaction so that the app executing on the VM is not harmed by the interference and information is not jeopardised. So, in order to avoid the loss of integrity and secrecy, we might concentrate on the data loss. Any weakness in network security, as any administrator knows, can lead in a data leakage and creative property; however, when it refers to VMM, which involves the entire operating system, any compromise to the network also can lead in a loss of VM monitor reliability. To summarise, protecting a virtualized network necessitates particular access control add-ons as well as a strategy that ensures complete separation from other network devices.

REFERENCES

1. Yiqiu F, Chen Z, Junwei G. Improvement on Live Migration of Virtual Machine by Limiting the Activity of CPU. In 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC) 2017 Dec 25 (pp. 1420-1424). IEEE.
2. Li H, Xiao G, Zhang Y, Gao P, Lu Q, Yao J. Adaptive live migration of virtual machines under limited network bandwidth. In Proceedings of the 17th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments 2021 Apr 16 (pp. 98-110).
3. Aldossary M, Djemame K. Performance and Energy-based Cost Prediction of Virtual Machines Live Migration in Clouds. In CLOSER 2018 Mar 19 (pp. 384-391).
4. Mazrekaj A, Nuza S, Zatriqi M, Alimehaj V. An overview of virtual machine live migration techniques. International Journal of Electrical & Computer Engineering (2088-8708). 2019 Oct 15;9(5).
5. Hines MR, Gopalan K. Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning. In Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments 2009 Mar 11 (pp. 51-60).

6. Zhang F, Liu G, Fu X, Yahyapour R. A survey on virtual machine migration: Challenges, techniques, and open issues. *IEEE Communications Surveys & Tutorials*. 2018 Jan 17;20(2):1206-43.
7. Fuentes E, Arce L, Salom J. A review of domestic hot water consumption profiles for application in systems and buildings energy performance analysis. *Renewable and Sustainable Energy Reviews*. 2018 Jan 1;81:1530-47.
8. Noshay M, Ibrahim A, Ali HA. Optimization of live virtual machine migration in cloud computing: A survey and future directions. *Journal of Network and Computer Applications*. 2018 May 15;110:1-0.
9. Karthikeyan K, Sunder R, Shankar K, Lakshmanaprabu SK, Vijayakumar V, Elhoseny M, Manogaran G. Energy consumption analysis of Virtual Machine migration in cloud using hybrid swarm optimization (ABC–BA). *The Journal of Supercomputing*. 2020 May;76(5):3374-90.
10. Lee CA, Zhang Z, Tu Y, Afanasyev A, Zhang L. Supporting virtual organizations using attribute-based encryption in named data networking. In 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC) 2018 Oct 18 (pp. 188-196). IEEE.
11. Khosravi A, Nadjaran Toosi A, Buyya R. Online virtual machine migration for renewable energy usage maximization in geographically distributed cloud data centers. *Concurrency and Computation: Practice and Experience*. 2017 Sep 25;29(18):e4125.
12. Bezerra P, Martins G, Gomes R, Cavalcante F, Costa A. Evaluating live virtual machine migration overhead on client's application perspective. In 2017 International Conference on Information Networking (ICOIN) 2017 Jan 11 (pp. 503-508). IEEE.
13. Basu D, Wang X, Hong Y, Chen H, Bressan S. Learn-as-you-go with megh: Efficient live migration of virtual machines. *IEEE Transactions on Parallel and Distributed Systems*. 2019 Jan 18;30(8):1786-801.

CHAPTER-3

[SECURITY ISSUES IN CLOUD COMPUTING AT THE VIRTUALIZATION LAYER]

The growing desire for more advanced technology and a more resourceful environment has given rise to cloud computing, which is extensively dispersed and offers on-demand, on-the-fly services. Cloud computing environments are vulnerable to a variety of threats due to their open nature and automated service provisioning. We have developed taxonomy in this study that lists several security vulnerabilities in cloud settings, particularly at the virtual machine. We tried to study the areas that demand greater effort for strengthening cloud virtualization privacy, along with a quick analysis of possible attacks on the cloud virtual machine and their present handling mechanisms.

3.1 SECURITY TECHNIQUE FOR CLOUD COMPUTING ON THE VIRTUAL SERVER

The cloud computing system gains crucial advantages like Rapid Ductility and Flexibility thanks to a large collection of integrated resources for service provisioning, but at the cost of increased potential threat. As services become more widely available, the chances of an assault increase. The cloud environment is a multi-tenant system in which virtual machines share software and hardware resources. The third-party cloud provider handles the data created, consumed, maintained, and discarded by cloud users, and the data is maintained in a central store called data centres.

Users can access applications, platforms, and infrastructure through cloud computing. Virtualization layers are used to deliver all of these functions. As a result, in addition to SaaS, PaaS, and IaaS, virtualization must be included while discussing secure services. Furthermore, because VM, Hypervisor (VMM), and Host System are all elements of the virtualization layer, any study of virtualization safety also includes VM, VMM, and Host Machine.

Taxonomy is described in Fig. 3.1, which contains criteria relevant to cloud privacy in general and focuses primarily on virtualization safety, as discussed above. The same includes security techniques for identifying and preventing the aforementioned concerns.

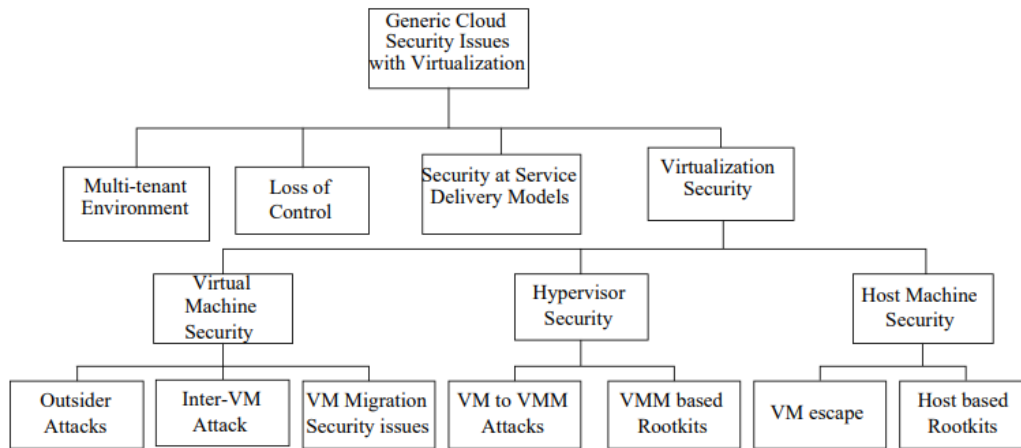


Figure 3.1 Cloud Computing Virtualization Security Taxonomy

3.1.1 Multi-tenant Situation

Access to numerous users can be offered concurrently in a cloud computing environment, from simple apps to huge software and many hardware resources. The idea behind a multi-tenant setup is that numerous virtual machines can share the same real system's resources. Virtualization hides the details of sharing resources from consumers, creating the impression that they are the only ones using that resource. Multi-tenancy has a lot of hazards, even though it works well because various people use the VMs. Stopping tenants from obtaining and overloading other tenants' resources is one of the most important characteristics to address for secure multi-tenancy.

Information sharing among renters is a significant entity to deal with, just like every other hardware & software resource allocation. A data centre is a centralised location where data and information are stored, managed, and disseminated. To support cloud capabilities such as service on need and flexibility, it is necessary to provide a virtual view of contiguous big memory amounts. A shared pool of storage resources is built in the data centre, and a large logical storage volume is constructed. Many security challenges, such as data security, data

separation, data verification, and authentication, arise when such a physically dispersed and theoretically united-contiguous storage architecture is shared across multiple autonomous clients.

Although cryptography can be used to safeguard data from unauthorised access, it may be broken if the key value is stolen. When information is deleted or encryption fails, it must be recovered. As a result, data centre security is an important consideration.

3.1.2. Control Loss

The Cloud System's services necessitate a changeover to a CSP. Because a third-party provider owns, handles, and controls user sensitive data and information in the cloud, the risk of data abuse, theft, and deletion increases. CSP's collaboration is required for operations administration and computing infrastructure choices. As a result, because the owner no longer has control over the resources, not only the selection of an appropriate CSP, but also the data architecture, should be done with care and attention to the security element.

3.2 ISSUES WITH SECURITY IN SERVICE DELIVERY SYSTEMS

The cloud system's services are divided into three categories: SaaS, PaaS, and IaaS . All of the services supplied to cloud customers are offered through one of these models, thus when we talk about cloud security, we need to talk about IaaS, SaaS, and PaaS [1].

3.2.1. Security at SaaS

Cloud Service Providers (CSPs) offer numerous software applications on the cloud infrastructure to users via SaaS. Clients can use these apps on their client gadgets without having to install or maintain software or care about underlying capabilities such as network, servers, operating systems, data storage, or adjustable application parameters. The SaaS approach is primarily concerned with providing consumers with application services. The applications' accessibility and availability may differ from one user to the next.

SaaS services include granting users varied file and program permissions, therefore identity administration becomes a critical component for ensuring permitted access.

3.2.2. Security at PaaS

PaaS concept provides an unified application development environment as well as a foundation with a variety of developed application frameworks, tools, and services. It provides the necessary operating system support as well as a platform on which to run an application. PaaS also includes the entire software innovation lifecycle, from strategy to architecture to execution to distribution to testing. PaaS security entails SOA-based protection and API security because cloud infrastructure is a SOA. All SOA security vulnerabilities, such as Man-in-the-Middle attacks, XML-related attacks, Replay attacks, Dictionary attacks, and Injection assaults, are also problems with PaaS security [2].

3.2.3. Security at IaaS

Infrastructure as a Service makes hardware and software resources available to clients on demand (IaaS). The CSP is the owner of the equipment and is responsible for housing, running, and supporting it in a virtualized environment. IaaS provides architecture, such as processing, storage, networking, operating systems, and other computational power, in response to user need. The basic function of IaaS is to integrate resources from several physical systems into a single cohesive format of required infrastructure, which is delivered as a virtual computer. Because IaaS manages virtual computers and customers access services through virtual servers, IaaS security was critical.

IaaS serves as the basis for SaaS and PaaS. According to [3,] the primary components of IaaS include Service Level Agreements, Cloud Services, Platform Architecture, Cloud Applications, Network & Web Access, and Computer Hardware, all of which must be addressed while enforcing IaaS security.

3.3. SECURITY IN VIRTUALIZATION

Virtualization enables cloud clients to request any hardware or software resource, such as a microprocessor, storage, disc, or system software, as a single logical unit made up of many disparate physical components. The environment's flexibility is increased by demand on a pay-per-use basis. This feature also causes a situation in which a large number of different types of VMs arise and vanish in the network in a short period of time. The software that runs in a virtual

world has a different lifecycle than software that runs in a traditional context. At the virtual layer, accessibility, heterogeneity, and authenticity are also factors to evaluate for security [4]. Because the network device, virtualization, and compute nodes are the most important elements of the virtualization layer, we'll go through the security challenges that they face in the sections below.

3.3.1. Security for VM

Because cloud customers access services using virtual machines (VMs), they are the most vulnerable to attacks from strangers or other VMs on the same given process. There is a risk of a VM being hacked even if it is moved to other physical machine. This section covers a variety of potential VM assaults.

3.3.1.1 Outsider Invasion

From a security standpoint, enabling a VM to make modifications to an existing system is crucial, yet certain apps need users to grant such permissions. To corrupt VMs and gain access to confidential information, users execute malicious apps. Although the notion of a captive account [5] may be used to limit customer rights, there are numerous additional methods, such as malware, worms, and misconfigurations that attack VM without compromising the allocated rights.

3.3.1.2 Inter-VM Assault

Virtual machines (VMs) are used to exchange computer hardware between different businesses. Multiple VMs can be run from the same theoretical model, even though they provide a virtual picture of a separate system. To support the idea of hybridization, the VMM provides perfect separation between VMs. Interaction between VMs is required for sharing resources and the deployment of particular applications, and it must be managed without jeopardising isolation. Inter-VM (cross-VM) attacks compromise this separation, causing virtualization to fail. Operations that necessitate data interchange must be permitted only after confirmation that the monitoring is carried out in accordance with a certain configuration and inside the permitted limits. A Cross-VM attack is also possible when authentication for common resource access fails. In addition, a technique known as a Side Channel Attack is used to leak classified data by

exploiting shared resources (SCA). SCA is a type of assault that is hard to detect since it steals sensitive data by monitoring metrics such as processing time, used power, and infrared photons to gain access to sensitive information. Cross-VM SCA is caused by a data flow assault between two virtual machines.

3.4. ISSUES WITH SECURITY DURING VM MIGRATION

Vm migration was the process of moving a virtual server from underlying hardware to another. Task scheduling, low latency, online maintenance costs, and virtual machine aggregation are all made easier with migration. VMs are halted or configured to function without disruption during the process of migration, which is referred to as route discovery. Although it aids in the maintenance of system state, task scheduling vulnerabilities provide a number of security risks. For a secure migration, examine a trusted source-destination, access control, secrecy, and integrity. [6].

3.5. SECURITY FOR HYPERVISORS

The main abstraction layer that separates virtual servers from the host system is the host or VMM. The virtualization level is broken when the hypervisor was damaged, and the resource concealment feature is degraded as a result. A brief review of parameters relating to hypervisor safety is presented here.

3.5.1 Attack from a VM to a VMM

The hypervisor gives virtual machine separation. The VMM must translate authorized orders or hypercalls produced by VMs into suitable system functions. Because the hypervisor separates VMs from the host computer, the abstraction layer falls down when the virtualization is attacked. Such attacks are carried out by VM-based apps that target VMM in order to take advantage of hybridization.

3.5.2 Rootkits at the VMM level

Rootkits are programmed to run at Ring-1, allowing them to take advantage of virtualization technology and monitor hardware calls performed by the original linux kernel. This type of rootkit could be installed on a computer's os before being promoted to a virtual machine. To

infiltrate the target, a virtualization rootkit does not need to make any changes to the kernel. Rootkit might be considered a key security factor to manage at the VMM level because it is difficult to identify its existence in the system.

3.6. SECURITY OF THE HOST MACHINE

The host system cannot be accessed immediately by customers in a virtualized environment since the user interaction is delivered solely through virtual machines. Despite this architecture, the host system has been targeted in the past by exploiting the intermediary layer that separates the virtual and physical levels. In this part, we'll go through some of the most common types of attacks.

3.6.1 Attack from a Guest to a Host

Guest applications must be able to run in a totally segregated environment with no access to tangible resources. Bugs in virtual servers can sometimes bypass the VM layer completely, allowing guest machines complete access to the host machine despite the existence of a hypervisor. The VM escape assault is when a guest computer attacks the host system.

3.6.2 Rootkits at the kernel level

Kernel-level rootkits are malicious programmes that are intended to get elevated access to a computer system and are difficult to detect unless a dedicated surveillance system is in place. Kernel-level rootkits operate at Ring-0 and change the BIOS booting process so that the rootkits are deployed before the bootstrap programme is executed. The identification of a kernel-level rootkit is difficult because the rootkit conceals its presence and so escapes detection by firewalls and other detection accuracy measures.

3.7. DoS

Excessive energy requests by a VM can result in a DDoS, starving other VMs on the same underlying hardware. Because of a lack of resources, critical processes may be hindered from being carried out. DoS attacks, if not properly implemented, can take the entire program down by overloading resources. According to [7], the hypervisor can be set to inhibit any VM from

having 100% access to services and could even reboot the VM if severe resource use is detected. DDoS attacks are DoS attacks that use numerous IP addresses to execute the attack.

3.8. SURVEY

3.8.1. Loss of Control and Multi-Tenancy

A data leak in a multi-tenant system, information security, and challenges resulting from a loss of control are all topics that many researchers are interested in. Security difficulties and challenges in the web, app, system software, virtualization, and hardware-software levels have been discussed by [8]. Each specified layer is divided into sorts of security issues unique to it, with work on each of them provided. Although [9] has taken into account multi-tenant environment challenges, there is still more work to be done in terms of data security. Segregation, Reliability, Mobility, Tunneling, and Encryption are among the 5 methods presented in [10] for dealing with various data security-related concerns. A data leak in a multi-tenant system, information security, and challenges resulting from a loss of control are all topics that many researchers are interested in. Security difficulties and challenges in the web, app, system software, virtualization, and hardware-software levels have been discussed by [8]. Each specified layer is divided into sorts of security issues unique to it, with work on each of them provided. Although [9] has taken into account micro environment challenges, there is still more work to be done in terms of data security. Segregation, Reliability, Mobility, Tunneling, and Encryption are among the 5 methods presented in [10] for dealing with various data safety concerns. [15] has developed a data auditing methodology for assuring data security as well as techniques for improving data validity. [16] took an alternative approach to data security, presenting an object tracking IDS for autonomous administration in a cloud environment. They proposed a technique for detecting data transformations, back propagation to reduce input size, and a method for detecting anomalous behaviour in automatic modes. According to the findings, several researchers have concentrated on various areas of data security such as data accuracy, data security, and flexible data support, which aids in the design of a safe multi-tenant system with a central database system. When dealing with challenges that arise as a result of a loss of control, it's important to think about authentic third-party network services and authentic communication services. The latter simply relates to the security of data centres. In [17], they

examine security challenges mostly on the memory and data levels, with a focus on improving third-party service safety. They talked about a system for secure third-party content distribution, as well as secure query processing with Map Reduce and Hadoop. [18] proposes a security framework for improving collaboration across cloud service providers, network operators, and service customers along the same lines.

3.8.2. Layers of Service Delivery Safety

We examined security vulnerabilities in the SaaS, PaaS, and IaaS models in section 3.3. Virtualization is discussed in the majority of the articles on IaaS. [19] describes IaaS intrusion prevention approaches that can distinguish ddos attacks emanating from individual VMs, even when numerous VMs share a single IP address. [20] focuses on the security issues that arise as a result of virtualization at the IaaS layer, concluding that fundamental security approaches propose security through data encryption or authentication protocols. He has gone over machine hypervisor, edge computing, and physical realm in great detail. [21] has provided a policy-based security method with a dynamic model that covers architecture scope for network, server, storage, and system management areas. [22] presented a Secure Model for IaaS (SMI), which uses a cubic model to define the interaction between IaaS Parts and related security criteria. With objects like SCP, SRMP, and SPMA, IaaS elements like Computing Services, Platform Automation, and Computer H/W are proven to fulfill security criteria. They argue that the aforementioned entities serve as a foundation for IaaS layer standardisation. Alharkan et al. present another IaaS-based flexible and movable signature-based IDS system that is totally managed by cloud consumers. [[23]]. Some approaches concentrate on SaaS or PaaS as well. [24] focused on cloud-based SaaS online services and presented an anomaly-based intrusion detection system (IDS) for collecting and evaluating large amounts of data without being attacked. The accuracy of the acquired data is also compared using different models. [25] has covered a variety of SaaS security topics, including data security, information security, data proximity, integrity of data, data separation, access to data, identification, and permission, as well as a brief look at PaaS and IaaS security. [26] cover XML Signature, SaaS level, PaaS, and IaaS.

3.9. SECURITY IN VIRTUALIZATION

Without a secure virtualization layer, safe SaaS, PaaS, IaaS, and Information Centers abstractly destroy the entire means of service providing. Virtualization security contributions range from analysing relevant security concerns to offering mitigation solutions for different host, VM, and VMM-based security assaults. In [13], the notion of virtualization and its different variants are discussed as a fundamental technique. It has covered potential virtualization risks such as VM Escape, DoS, and security risks between VMs and VMM. A full overview of how to deal with them using hardware and software is also included.

[28] developed a platform for virtual server security, network virtualization information security, and policy-based trust consulting services. [29] has highlighted a set of security concerns for cloud computing emulation environments such as Role-based Network Access, Data Separation, and Customer Confidentiality, while [30] presents security needs, attacks, and security systems for virtualization. [31] has done a thorough review of the virtualization consequences in the literature. To maintain data security, they recommend removing or carefully managing non-essential functions such as reflection, backup, transmission and security for cloud computing, with an emphasis on processor security. As we covered in section 4, hypervisor layer assaults can be divided into three categories: VM, VMM, and host device. In the subsections that follow, diagnostic methods to these threats are described.

3.10. SECURITY FOR VM

A virtual machine (VM) can be exploited by another VM, the host machine, or an external attacker. As a result, the effort done to address various aspects of VM safety can be summarised as follows.

3.10.1 Outsider Invasion

A user running numerous vulnerable applications on a virtual machine is a big vector of attacks. [32] has provided a strategy that monitors VM activities with a sandbox approach and a Host Level-Guest Levels Of security Analyzer, which is one of the methods suggested for tracking of Outsider threats. The software monitoring tasks are split between the virtual machine, the virtual machine manager, and the host device. [33] offered a solution with effective surveillance ability

in another method with a decentralized security approach. This method creates a hook inside the core of the virtual machine to detect malware and redirect it to a specialised virtual machine for analysis. Unlike passive surveillance, which can only be done from a protected VM with wireless monitoring, this method is active. Both of the methods described above can prevent malware from entering a VM while imposing a significant amount of space overhead. Suspicious activity can be detected by monitoring traffic flow at a fine detailed level . For a small number of VMs, this strategy provides low-level performance overhead. However, because VMM manages the security system IDS, as the number of VMs grows, so does the load on VMM. Other techniques to Hypervisor orientation are security bottlenecks. Such VMM-centric techniques exist. Utilizes the smart disc technique and has created a storage-based IDS that allows the virtual disc to keep track of the sector-to-file mapping database. To identify the attack, ID features are included into digital storage. The method of [34] identifies harmful actions by adjusting dynamically in response to the number and status of virtual machines (VMs) and gathering data on file alteration and stability. [35] have proposes an algorithm that attributes and implements exploitable flaws predicates for positioning thresholds to monitor the validity of the target systems (VMs) at different points and with new and old time frames, and offers spots to manage them, observing in the footsteps of VMM-based security mechanisms. A non-recursive method that utilizes the host view casting process to rebuild a limitations view of VM externally. With enhanced accuracy rate, the suggested method verifies the validity using 2 methods: view comparison-based virus identification and host-based anti-malware technology. Some methods keep a committed VM to identify intruders when the previous techniques use Hypervisor. [35] uses software reverse proxy, inter-VM disc installation, and inter-VM activity mapping to protect each host VM from assaults without any need for extra hardware. [36], on the other hand, keeps a dynamic rule list updated as new VMs are created and old ones are deleted. It takes advantage of VMM's separation property to detect real threats by comparing them to a regularly updated rule set and cross-domain interaction. Both strategies have a low overhead in terms of time. [37] proposes a powerful domain firewall that keeps a white-listed process list on the elevated domain and verifies the connection legitimacy for each request. It can restore purpose after VM by replaying the precise execution process of even probabilistic VM apps. [38] proposes a host-level VM-logging approach, in which a surveillance device

independent from the host identifies invasions by tracking the victim VM even when the machine's core is updated. It can repeat the precise iterative sequence of even non - deterministic VM programmes in order to restore functionality once the VM has been compromised. [39] have proposed a DDoS-targeting strategy based on a paravirtualized layer. They estimate the following secure host operating network architecture with a PV-Basement layer beneath the virtual machine. With VM-Shadow and Program Detector, it detects and manages VMs of comparable types from app views. This approach necessitates a significant amount of overhead space. The work done to manage outsider attacks demonstrates that when an end user runs a virtual machine remotely, it becomes exposed to a variety of attacks. All of the above-mentioned techniques have a sustainability overhead, as seen in Table 1.

Table 1 Extrovert Attack on a VM

Author	Position of Attack Handling System	Scalability Overhead
M. Noura et al. [53]	Distributed	High level Space Overhead
Bryan D. Payne et al. [54]	Distributed	High Level Space Overhead
U. Tapukala et al [55]	VMM level	Low Level Time Overhead
F. Zhao et al. [61]	Dedicated VM	High Level Time and Space Overhead
Y. Zhang et al. [56]	VMM Level	Low Level Time Overhead
T. Garfinkel et al. [17]	HIDS, separated from Host	Low Level Time Overhead
G. Dunlap et al. [63]	Host Based VM-Logging System	Low Level Time Overhead
H. Jin [57]	Hypervisor Level	Low Level Time and Space Overhead
K. Kourai et al. [60]	Privileged VM	Low Level Time Overhead
Joshi et al. [58]	Hypervisor based IDS	Low Level Time Overhead
Jiang X. et al. [59]	Hypervisor based IDS	Low Level Time Overhead
A. Srivastava et al. [62]	Privileged domain Firewall	Low Level Time Overhead
D. Patidar et al. [64]	Works on Para Virtualization layer to target DDoS attacks	High Level Space Overhead

3.10.2 Inter-VM Assault

Multiple VMs are considered to be co-resident when they share computer equipment. Classifier model is shared among these virtual machines. [40] provided a method for deploying an attacker VM on the same given system as the target VM. The offender opens the door to several types of vulnerabilities with a similar access point by making the rogue VM co-resident with the victim VM. This results in a Cross-VM attack. [41] et al. introduced a novel approach, PSSF to distribute VMs with the goal of reducing the likelihood of co-resident assaults. For optimal behaviour, workload management and energy consumption are also taken into account. The system works well as long as the server processing is done according to the assumptions made in this technique. Lefray et al. have presented a new metric for sustaining isolation features in [42],

which focuses on a covert channel based information 21 leaking. Table 2 shows the sustainability overhead of techniques for dealing with SCA on secondary storage.

Table 2 Approaches to Defending Against Cross-VM Attacks

Author	Focused Parameter	Scalability Overhead
B. Sevak [84]	Side Channel Attack	Time Overhead
S. Yu [87]	Cache Side Channel Attack	Low Time Overhead
B. K. Mughal [85]	Cache Side Channel Attack	Time Overhead
M. Chouhan [86]	Cache Side Channel Attack	Time Overhead
Fangfei Liu [88]	Handling of SCA to Last Level Cache Memory	Limited Scalability
Fangfei Liu [89]	Handling of SCA to Cache Memory	Negligible
Yi Han et al. [90]	Defending against Co-residency	Negligible
Lefray et al. [91]	Cache Leakage	Low Time Overhead

3.11 ISSUES WITH SECURITY DURING VM MIGRATION

Live VM Migration necessitates procedures that ensure state data is secure even if the data structure is in progress. [43] discusses the parameters for facilitating live relocation and how they are met with certain existing information security. They took into account factors such as the CoM framework, the Virtual Trusted Platform, the Live Migration Defense Framework, and role-based policies. Many experts have worked on developing reliable VM migration techniques. [44] conducted a survey on virtual machine transition security challenges and solutions. In this review, which compares existing security techniques to preserve integrity, identification, and secrecy during Virtual machines, threats such as unprotected data transmission, improper user access rules, and relocation unit loopholes are examined. [45] uses a policy-based approach to secure live VM migration, using Attestation Server and Sealed Memory to safeguard VM migration with encryption. [46] proposes energy-aware procurement of cloud computing services in virtualized systems as a very practical approach for TOCTTOU, VM Initiation Scheduling, and Playback assaults in another way for live migration. As the rate of VM movement rises, both options incur overhead. [47] has explored the security challenges that arise during vm migration in terms of 3 planes: management, information, and migration unit. Also presented is a tool for automating VM memory management, as well as solutions for addressing VM design flaws. Only Xen and VMWare VMM security vulnerabilities are discussed in this method.

Table 3 Security Threats During VM Migration

Author	Reliability
Wei Wang et al. [100]	Good
Mahdi Aiash et al. [99]	Good
Korir Sammy et al. [101]	Good
Jon Oberheide et al. [102]	High
Rajesaheb R. Kadam et al. [19]	Good
G. Jarlin Jeincy et al. [103]	Good but Considerable Time Overhead
P.Jabalin Reeba [104]	Good with Low Time and Space Overhead

3.11.1. Hypervisor Security

The software layer that divides physical and virtual systems is known as the hypervisor. For safe virtual services, assaults on the hypervisor must be identified and stopped. VMs and rootkits are the most common threats to a VMM.

3.11.1.1 VM to VMM

Attack A hostile virtual machine can undermine the hypervisor, which is at the heart of the virtualization layer. There are just a few methods for dealing with guest-to-hypervisor attacks. [48] has proposed a Collabra approach as one of them. Collabra is a VMM-integrated IDS that uses its elevated Dom 0 hosting zone to filter fraudulent hyper calls and may have many hosting examples for each VMM working together to discover affected hypercalls. Another method is to use [49] to handle VM to VMM attacks. offer a VMM-based model with Behavior Processing and Evaluation elements that evaluates system call behaviour to normal behaviour and discovers abnormalities finding divergence. Techniques for assessing VM to VMM attacks are placed on VMM, thus as the number of VMs grows, so does the chance of an attack, and hence the burden on VMM.

Table 4 shows the manageability overhead of the tactics outlined above, as well as the mechanism used to execute the attack.

Table 4 Methods for dealing with a VM to VMM attack

Author	Method Employed	Scalability Overhead
S. Bharadwaja et al. [105]	Scanning for Hyper-calls for integrity checking	Low Level Space Overhead
Y. Du et al. [106]	Scanning System Calls from privileged programs	Low Level Space Overhead

3.12. HOST MACHINE SECURITY

Because a hacked host machine impacts every element of the virtualization layer, it must be fully error-free. VMM bypassing VMs or kernel-level rootkits could both be used to attack it.

3.12.1 Guest-to-Host Attack

The VM Escape assault overcomes the hypervisor and causes the VM to immediately exploit the host. Cloudburst is described as a VM Escape attack for KVM and VMWare, however no practical real-time implementation has yet been discovered. According to the report, quite an attack could be carried out by inserting bind shell code into the Dom 0 root process, allowing it to directly access storage. The attack can either be carried out via raising hypervisor capability or declining calling zone privilege. Approaches to ensuring hypervisor security are developed in the hypervisor is evaded to make VM interface with the hardware by applying strategies such as pre-allocation of resources, customised guest OS, and use of virtualized I/O to deal with VM Escape. The removal of the virtualization also reduces the necessity of this attack, but it is difficult to accomplish in practise, and the overhead of the constantly interacting as VMs are dynamically created and destroyed. As a result, some mechanism is implemented to avoid VM Escape with the least amount of complexity and without altering the hypervisor secure communication. Table 5 shows the adaptability overhead of the previously stated techniques.

Table 5 Rootkits at the kernel level

Authors	Scalability Overhead
Nick L. Petroni et al. [115]	Negligible
Ryan Riley et al. [116]	Space Overhead
Lionel Litty et al. [118]	Time Overhead
Arvind Seshadri et al. [117]	Space Overhead
Abhinav Srivastava et al. [119]	Time Overhead
S. Jones et al. [120]	Time Overhead
M. Ficco et al. [121]	Low Time Overhead
R. Hund et al. [122]	Low Time Overhead
Z. Wang [123]	Time Overhead

3.13. DENIAL OF SERVICE ATTACK

The most frequent types of DoS threats in a private cloud are flooding and IP spoofing. TCP SYN Flood Attacks from one VM to another was investigated and tested with various set parameters. Once they do not react to SYNC/ACK, an enabled IDS is configured to log all incoming-outgoing traffic and request a ping location. When no response to a permission slip is received, DDoS is suspected. To counteract the attack, a virtual machine switch is implemented. There are numerous publications that examine techniques to dealing with DoS and DDoS assaults in a cloud context; this one approach is taken by Opeyemi, who has conducted an extensive literature review on reducing DDoS attacks.

3.13.1 Security model/architecture at Virtualization Layer

In addition to the methodologies stated above, the researchers have offered some model types that are created with security in mind. VMM was suggested with the capabilities of detecting guest-middleware stability and intuitively shielding them from the majority of threats..

In, a model for offering safe virtualization was offered that discussed cloud security challenges and presented ACPS. With full visibility to VMs and cloud customers, the proposed approach can monitor the quality of guest computers and equipment. Has developed an SVL paradigm for dealing with security issues at the virtual machine. Splitting the virtualization into smaller modules is a notion. The goal of the split virtualization is to lower the size of the TCB in order to make hypervisor system security easier. The suggested security features necessitate a change to the existing virtualization code, which increases complexity and increases the overall exposure of the Network Architecture Base. As a result, due to its code surface, protecting the

virtualization becomes more critical and difficult. Table 6 shows the overhead imposed by the suggested security infrastructure on manageability. Table 6: Privacy Architectures for the Virtual Machine Proposed.

Authors	Scalability Overhead
A. Volokyta et al. [127]	Low Level Time and High-Level Space Overhead
F. Lombardi et al. [128]	Low Level Time Overhead
S. Manavi et al. [129]	Low Level Time Overhead
W. Pan et al. [130]	Low Level Time Overhead

3.14 SUMMARY

As the number of applications for cloud computing grows, so does the necessity for highly safe services. At each level of cloud environment, different types of security risks must be addressed. Installing a firewall, IDS/IPS, and/or encryption techniques can enforce safety at cloud data stores, diverse service delivery methods, and control, according to the survey. Because of the types of security issues that are likely to be overwhelmed with conceptual services, effective use of such processes can create the public cloud secure. However, security problems pertaining to vms necessitate more attention based on the types of security problems that are likely to be raised with esoteric services. The analysis of work done in dealing with various assaults on virtualization reveals that there is a lot of research being done in dealing with malware (outsider) assaults on VMs as well as kernel-level and hypervisor-level rootkits. There are various approaches that focus on the VM-VMM attack and security vulnerabilities that arise during VM migration. Although the presented methodologies are secure against their individual vulnerabilities, they increase overhead when sustainability is taken into account. Furthermore, techniques with low-level overhead have other constraints, so security flaws continue to exist in the system.

REFERENCES

1. Singh S, Jeong YS, Park JH. A survey on cloud computing security: Issues, threats, and solutions. Journal of Network and Computer Applications. 2016 Nov 1;75:200-22.

2. Khan MA. A survey of security issues for cloud computing. *Journal of network and computer applications*. 2016 Aug 1;71:11-29.
3. Coppolino L, D'Antonio S, Mazzeo G, Romano L. Cloud security: Emerging threats and current solutions. *Computers & Electrical Engineering*. 2017 Apr 1;59:126-40.
4. Ali M, Khan SU, Vasilakos AV. Security in cloud computing: Opportunities and challenges. *Information sciences*. 2015 Jun 1;305:357-83.
5. Modi CN, Acha K. Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review. *the Journal of Supercomputing*. 2017 Mar 1;73(3):1192-234.
6. Sgandurra D, Lupu E. Evolution of attacks, threat models, and solutions for virtualized systems. *ACM Computing Surveys (CSUR)*. 2016 Feb 8;48(3):1-38.
7. Iqbal S, Kiah ML, Dhaghighi B, Hussain M, Khan S, Khan MK, Choo KK. On cloud security attacks: A taxonomy and intrusion detection and prevention as a service. *Journal of Network and Computer Applications*. 2016 Oct 1;74:98-120.
8. Han Y, Chan J, Alpcan T, Leckie C. Using virtual machine allocation policies to defend against co-resident attacks in cloud computing. *IEEE Transactions on Dependable and Secure Computing*. 2015 May 4;14(1):95-108.
9. Lefray A, Caron E, Rouzaud-Cornabas J, Toinard C. Microarchitecture-aware virtual machine placement under information leakage constraints. In *2015 IEEE 8th International Conference on Cloud Computing 2015 Jun 27 (pp. 588-595)*. IEEE.
10. Hussein O, Hamza N, Hefny H. A proposed covert channel based on memory reclamation. In *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS) 2015 Dec 12 (pp. 343-347)*. IEEE.
11. Tan Y, Wei J, Guo W. The micro-architectural support countermeasures against the branch prediction analysis attack. In *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications 2014 Sep 24 (pp. 276-283)*. IEEE.
12. Zhang L, Shetty S, Liu P, Jing J. Rootkitdet: Practical end-to-end defense against kernel rootkits in a cloud environment. In *European Symposium on Research in Computer Security 2014 Sep 6 (pp. 475-493)*. Springer, Cham.

13. Osanaiye O, Choo KK, Dlodlo M. Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework. *Journal of Network and Computer Applications*. 2016 May 1;67:147-65.
14. Orman H. Both Sides Now: Thinking about Cloud Security. *IEEE Internet Computing*. 2016 Jan 1;20(01):83-7.
15. Chouhan M, Hasbullah H. Adaptive detection technique for Cache-based Side Channel Attack using Bloom Filter for secure cloud. In 2016 3rd International Conference on Computer and Information Sciences (ICCOINS) 2016 Aug 15 (pp. 293-297). IEEE.
16. Liu F, Ge Q, Yarom Y, Mckeen F, Rozas C, Heiser G, Lee RB. Catalyst: Defeating last-level cache side channel attacks in cloud computing. In 2016 IEEE international symposium on high performance computer architecture (HPCA) 2016 Mar 12 (pp. 406-418). IEEE.
17. Liu F, Wu H, Mai K, Lee RB. Newcache: Secure cache architecture thwarting cache side-channel attacks. *IEEE Micro*. 2016 Oct 27;36(5):8-16.
18. Jeincy GJ, Shaji RS, Jayan JP. A secure virtual machine migration using memory space prediction for cloud computing. In 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT) 2016 Mar 18 (pp. 1-5). IEEE.
19. Reeba PJ, Shaji RS, Jayan JP. A secure virtual machine migration using processor workload prediction method for cloud environment. In 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT) 2016 Mar 18 (pp. 1-6). IEEE.
20. Almorsy M, Grundy J, Müller I. An analysis of the cloud computing security problem. *arXiv preprint arXiv:1609.01107*. 2016 Sep 5.
21. Wani AR, Rana QP, Pandey N. Analysis and countermeasures for security and privacy issues in cloud computing. In *System performance and management analytics 2019* (pp. 47-54). Springer, Singapore.
22. Wu Y, Liu Y, Liu R, Chen H, Zang B, Guan H. Comprehensive VM protection against untrusted hypervisor through retrofitted AMD memory encryption. In 2018 IEEE International Symposium on High Performance Computer Architecture (HPCA) 2018 Feb 24 (pp. 441-453). IEEE.

23. Srinivas TM, Amritha PP. Real Time Audio Steganographic Countermeasure. In *Data Engineering and Intelligent Computing 2018* (pp. 293-300). Springer, Singapore.
24. Wani AI, Lone ZA. A Survey of security issues and attacks in cloud and their possible defenses. *International Journal of Emerging Technologies in Engineering Research*. 2017 Dec;5(12):97-109.
25. Li S, Xue M, Zhao B, Zhu H, Zhang X. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*. 2020 Sep 3.
26. Kofahi NA, Al-Rabadi AR. Identifying the top threats in cloud computing and its suggested solutions: a survey. *Adv. Netw.* 2018 Mar 20;6(1):1-3.
27. Kumar R, Goyal R. Top Threats to Cloud: A Three-Dimensional Model of Cloud Security Assurance. In *Computer Networks and Inventive Communication Technologies 2021* (pp. 683-705). Springer, Singapore.
28. Amara N, Zhiqui H, Ali A. Cloud computing security threats and attacks with their mitigation techniques. In *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)* 2017 Oct 12 (pp. 244-251). IEEE.
29. Ahmad N. Cloud computing: Technology, security issues and solutions. In *2017 2nd International Conference on Anti-Cyber Crimes (ICACC)* 2017 Mar 26 (pp. 30-35). IEEE.
30. Pekaric I, Sauerwein C, Haselwanter S, Felderer M. A taxonomy of attack mechanisms in the automotive domain. *Computer Standards & Interfaces*. 2021 Apr 23:103539.
31. Akshaya MS, Padmavathi G. Taxonomy of security attacks and risk assessment of cloud computing. In *Advances in big data and cloud computing 2019* (pp. 37-59). Springer, Singapore.
32. Qayyum A, Ijaz A, Usama M, Iqbal W, Qadir J, Elkhatib Y, Al-Fuqaha A. Securing Machine Learning in the Cloud: A Systematic Review of Cloud Machine Learning Security. *Frontiers in big Data*. 2020;3.
33. Anand P, Singh Y, Selwal A, Singh PK, Felseghi RA, Raboaca MS. IoVT: internet of vulnerable things? Threat architecture, attack surfaces, and vulnerabilities in Internet of Things And Its Applications Towards Smart Grids. *Energies*. 2020 Jan;13(18):4813.

34. Kaur U, Mahajan M, Singh D. Attacks Surfaces and Attacks in Cloud Computing. *Journal of Operating Systems Development & Trends*. 2019 Jan 16;5(3):6-9.
35. Chow MC, Ma M, Pan Z. Attack Models and Countermeasures for Autonomous Vehicles. In *Intelligent Technologies for Internet of Vehicles 2021* (pp. 375-401). Springer, Cham.
36. Isakov M, Gadepally V, Gettings KM, Kinsy MA. Survey of attacks and defenses on edge-deployed neural networks. In *2019 IEEE High Performance Extreme Computing Conference (HPEC) 2019 Sep 24* (pp. 1-8). IEEE.
37. Ranjan A, Selvaraj R, Kuthadi VM, Marwala T. Stealthy Attacks in MANET to Detect and Counter Measure by Ant Colony Optimization. In *Advances in Electronics, Communication and Computing 2018* (pp. 591-603). Springer, Singapore.
38. Babu DS, Reddy PC. Prevention of Stealthy Attacks through Privacy Mechanism in Wireless Ad hoc Networks. *Indian Journal of Science and Technology*. 2017 Apr 25;10(16).
39. Varatharajan R, Preethi AP, Manogaran G, Kumar PM, Sundarasekar R. Stealthy attack detection in multi-channel multi-radio wireless networks. *Multimedia tools and applications*. 2018 Jul;77(14):18503-26.
40. Muruganandam D, Manickam J. Detection and Countermeasure of Packet Misrouting in Wireless Adhoc Networks. *Sensor Letters*. 2019 Sep 1;17(9):696-700.
41. Latha SP, Sabitha R, Anitha K, Nalini M. A Novel Technique for Jamming Attack Detection in Wireless Adhoc Networks Using BLMSPC Protocol. *Annals of the Romanian Society for Cell Biology*. 2021 Apr 11:2766-74.
42. Muruganandam D, Manickam JM. An efficient technique for mitigating stealthy attacks using MNDA in MANET. *Neural Computing and Applications*. 2019 Jan;31(1):15-22.
43. Zougagh H, Idboufker N, Zoubairi R, El Ayachi R. Prevention of Black Hole Attacks on Mobile Ad Hoc Networks Through Intrusion Detection Systems. *International Journal of Business Data Communications and Networking (IJBDCN)*. 2019 Jul 1;15(2):73-91.

CHAPTER-4

SOFTWARE DEFINED NETWORK PROCESS FOR OVERCOMING ATTACKS IN VIRTUALIZATION

The separated promotional devices provide the Software Defined Network (SDN) system operator with a lot of advantages. The SDN will address security concerns as well as the community's significant liabilities. The most serious risk is a DDoS assault. The purpose of this research is to recommend a learning strategy for DDoS attacks using an SDN-based system. Disturb the user's legal acts to recommend ALM as a more advanced set of SVM to return specific viabilities. Two forms of flooding-based DDoS assaults are identified in this study. The separated promotional devices provide the SDN system controller with a lot of advantages. Using the main elements, such as volumetric and asymmetric properties, the suggested Virtualization approach reduces exercise and testing time. The revealing technique is roughly 97 percent accurate in terms of fastest practise and investigation duration.

4.1 SERVICE ATTACKS

SDN is a new network architecture in which network control is dynamic, configurable, adaptable, and physically separate from accelerating devices [1]. Reliability, adaptability, resilience, and compatibility are the primary challenges of SDN. Each layer of SDN has susceptibilities, according to the focus focused on SDN security. Single network unit switches are quite sensitive to various types of attacks on service providers, like DoS attacks, side channel attacks, DDoS attacks, black hole attacks, negation, and data manipulation, at the information level.

DDoS and DoS are common attacks on the channel's data layer, which are inaccessible to legitimate users. Because the initial packet of each stream must be transmitted to the operator, the data plane is the safest choice for DDoS, and it can occasionally cause a slowdown. At the control plane, cyber threats such as DoS, black hole, and phoney flow control creation can

happen. Some vulnerabilities in the app plane related to DDoS attacks are being examined in Smart City applications.

4.2 SDN-BASED DETECTION OF SERVICE ATTACKS

ASVM approach is used to identify DDoS threats on the SDN channel. The proposed study introduces a configurable DDoS security structure that provides DDoS attack notifications based on the app's security requirements [2]. As a result, the proposed work was energised by the idea that various claims demand distinct security criteria.

The suggested context takes into account the unpredictability of DDoS attacks and the necessity for a tailored alert system for creating DDoS attacks. As such, the handler mechanism proposes a dynamic DDoS defence NFV technique that utilises system architecture with active equipment and environment. DDoS attacks are simple to launch but difficult to defend against. A botnet is a collection of computers that is used to launch DDoS assaults. DDoS threats are frequently classified into types depending on the directed standards platform [3] for the purpose of safeguarding the service operating security:

4.3. DDOS FLOODING ATTACKS ON THE NETWORK

These reductions frightened the massive treatment of DNS, ICMP, TCP, and UDP, protocol packets, specialising in interrupting genuine people's communication from end to end across the constrained network's throughput.

4.3.1 DDoS flooding attacks at the application level

Proposing open patrons' capacities to access the server attributes (– for example, ports, CPU, memory, sd card bandwidth, and I/O connectivity) are the focus of individual assaults. Portable devices such as phones and ipads have developed a reputation for being a critical production mindset for DDoS threats against cloud services. The lack of mobile safety for the general public, combined with increasing bandwidth and computation.

On the route to concern, power provides a fertile ground for hackers. In 2013, researchers discovered Android malware that could be used to commit DDoS assaults [4]. Cruel intruders now have a strong physical assault tool within the rave, and to use it, they need to employ the insignificant skill requests.

4.4 DEALINGS WITH A NEW BREED OF FIRMNESS AND AMENITY

Immediate opposition and dignified service adopters of cloud amenities continue to be paid on a need basis, while fog associated link properties are transformed using a traditional model DDoS depending on connected resources. In a newly updated cloud scenario, a new species of intruder assaults targets using cloud user financial shares. The package content, along with the destination node, origin IP address, and terminal port, will be used as ornamental material in the link layer fields.

Incoming packet data is compared to flow entries, and if a connection is found, a defined action can be performed. Instead, a packet in management packet will be delivered to the Uncluttered Daytime controller through the southbound API. The controllers are linked together in a cluster. When traffic arrives at the Exposed Dawn management cluster, it is sent through the northbound API to the suggested application level technique to detect DDoS attacks. Self-control is classified as either a DDoS spasm conveyance or a normal delivery. The suggested structure is made up of modules such as a mobility social group, a flow of traffic info gathering for extracting features, and attack identification.

4.4.1 Peer group for transportation

In this effort, the batch of twofold DDoS event traffics and regular traffics is realised. UDP saturating assaults and SYN inundating attacks are two types of DDoS attacks. UDP flooding is a form of DoS attack in which random ports on the object's interface are flooded with UDP. The IP addresses of the victims of a UDP flooding assault are identified, and the origin and target ports are reset to 80 and 1, respectively. A total of 2000 containers are created at any given moment. For UDP assault traffics, the packets bury access time is 0.04 seconds. Scapy, a

python-based packet social group program for microprocessor systems, was utilised to create the packets in this study.

Skimming, smidgen navigation, piercing, unit tests, convulsions, and linkage finding are all jobs that Scapy may switch between. The packet must be transmitted to the destination IP address inside the time interval after it has been produced. Figure 4.1 shows the step-by-step approach for launching a UDP overloading attack on an SDN network. SYN flooding was a sort of DoS attack that takes advantage of the normal three-way handshake mechanism to devour the contents of the damaged server and leave it pokerfaced by leveraging the TCP structure. For normal traffic flow production, the packet inter transit time is 0.4 second. Every time, the unintentional base IP address is utilised.

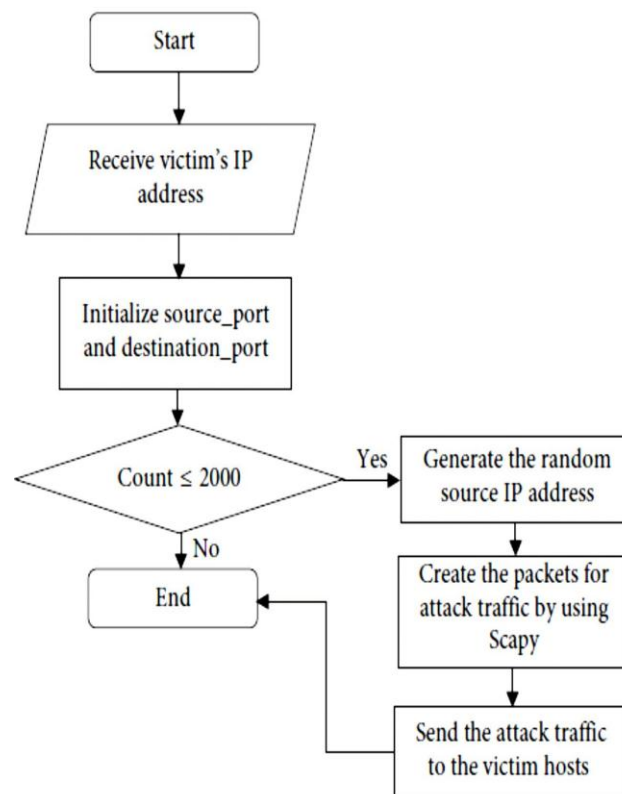


Figure 4.1. UDP Flooding Spasm Process in Steps.

4.5 DATA COLLECTION ON TRAFFIC

The traffic statistics are kept in the routing table in SDN, and the Public Course control reacts to the open flow stats request connection and sporadically directs this information to the organiser to retrieve the traffic information. Most DDoS attack mitigation instruments require data to be gathered in order to create a typical summary or identify abnormalities. In cloud environments, DDoS threats have grown in magnitude, making it more complex to collect fantastic and disparate facts with a squat overhead.

Furthermore, the facts of cloud transportation are distributed between network devices via content similarity, and the multi-tenant setting of fog environments makes data collecting for preventing DDoS attacks more difficult to achieve. Several intellectual processes, such as artificial neural networks, chaotic analysis, Bayesian cataloguing, game concept, HSMM, and fuzzy logic, have been employed to improve hindrance in cloud atmospheres. Due to the complexity of DDoS attacks, no single intellectual method can control all types of DDoS attacks. Selecting various brainy systems that result in changed spasms is a difficult task to tackle.

4.6 THE CLOUD EDGE TO CORE MODEL

The depth of the neural networks, as well as the quantity and reliability of the retraining data determine the validity of the models and their ability to evaluate fresh input. An end-to-end analysis of the rate at which DL datasets produce can be startling. The data passes through three stages in a DL model dispersion: edge (data ingest), center (learning groups, information lake), and cloud (data archival).

This data effort is the same in displays like IoT data, which spans 3 components of the communications network. The phases of the network system are depicted in Figure 4.2. The cloud can be used in a variety of ways, including GPU graphics for reckoning and data stores tiering archives and backups. The information in many Ai systems may be spread across the periphery, center, and internet. As the data is orchestrated throughout different settings, the adaptive approach outlined above, along with a current machine learning technique, can make a choice about extending or contracting networks.

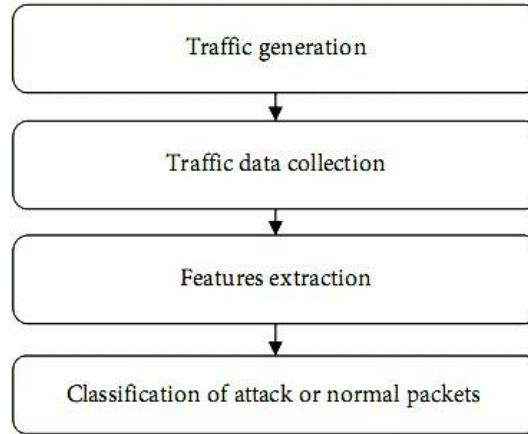


Figure 4.2. Modules of proposed system framework

The big guns are enigmatic, and they used to think of the process as unsupervised training. The training set for the Heavy Hitter Identification machine learning comprises the flow data, which denotes two causal statuses in the present situation.

$$\omega = \begin{cases} 1, & \text{heavy hitter network state} \\ 0, & \text{normal network activity} \end{cases}$$

The continuous improvement of cloud capabilities with SDN in Mobile Clouds (SDMC) is aimed at future unified mobile network research and technology. The intelligent services coordination and dynamic strategic planning capabilities of SDN and NFV [5] have been merged. SDN strives to decouple the networks' mechanisms starting with the data planes on its own. The intelligently combined intellect network influences on SDNs, as shown in figure 4.3, manipulates entire capabilities via the generalisation of key network architecture.

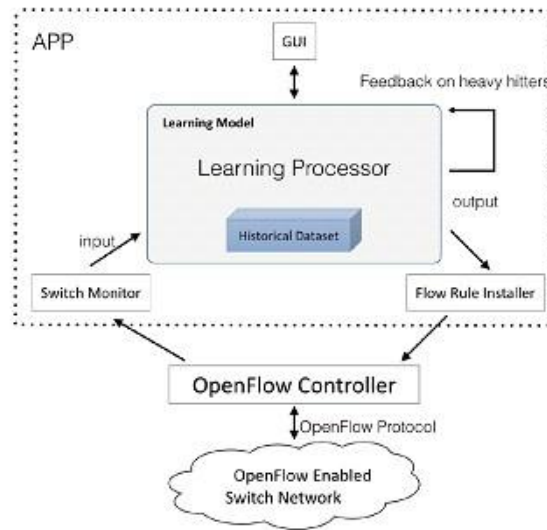


Figure 4.3. Algorithm design of a heavy hitter

In addition of various lowering scalability strain at low-cost, SDN distributes signals as suggestions to control systems. For packet port providers, flexible traffic management is possible, and scalability may be greatly improved with real-time updates. The fine-grained packet handling commands on forwarding status at the user level result in joint operator flexibility and rapid state conversion to avoid transporter interruptions.

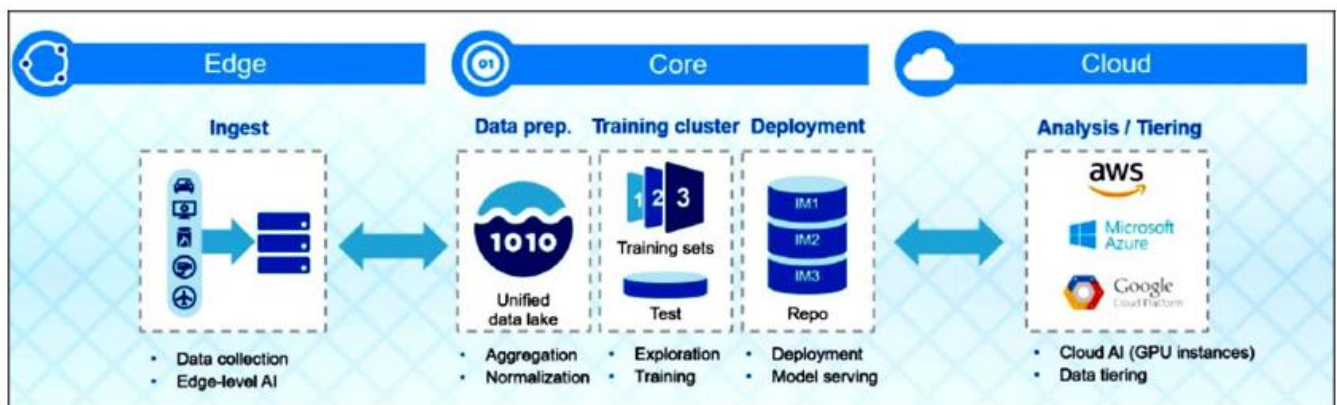


Figure 4.4 Cloud Model from the Periphery to the Core

With the help of changes, fine-grained traffic quantity tracking ensures that subscribers are alerted when they exceed their preset restrictions. Flexible user policies address bits of delaying problems, and QoS machine approaches should shorten statistics extend or transform at ease to

concede highest package offer via bits of waiting concerns. The massive increase of records, visitors, and connected nodes necessitates the use of such systems.

The development of expertise on SDN network feature virtualization (NFV) and Cloud Presuming concepts is advancing to meet the requirements of future cell networks. The positive effects of SDN mechanism regime include a new system, career placement, and community structure evolution, and those payments should invariably follow in distinctive interacting scenarios such as datacenters, wireless networks, broadband access networks, and campus networks. Also, in special networking contexts such as datacenters, wireless networks, wireless broadband networks, and campus systems, these rates of pay could be extremely valuable.

This technology provides services that are tailored to the needs of the consumers and may be grown while maintaining cheap prices. As a result, the cloud computing platform is being used by an increasing number of businesses. However, this rapid shift to the cloud has raised security concerns, as cloud computing has introduced new threats and issues. By analysing several dissimilar criteria of the current counter, the assembled malicious transportation actions on the SDN network container must be analysed.

4.7 RESULTS OF THE EVALUATION

On an Ubuntu 16.04 VMware, the portraits' enquiries are confirmed using the Mininet (version 2.3.0d1) emulator and the procedure to construct the SDN system topology. Mininet was a network emulator that operates a number of hosts, controllers, routers, and connects on a single Linux kernel, simulating the impacts of a genuine community [6]. Most DDoS assaults employ at least three hosts, with the number of hosts ranging from one to as many as one hundred, and the number of controllers used ranging from one to as many as possible.

One hundred hosts (h1 to h100), nine switches (s1 to s9), and three controllers are proposed as part of the SDN check mattress mechanism (c0, c1, c2). Our test mattress is divided into four subnets. Miniedit is where the experiments are set up. Miniedit was a simple Mininet GUI editor. Fig. 5 depicts our actual test bed data.

The mobility peer group was established in every circumstance, and then the traffic stream slide data from each control is actually developed or after each switch. Five unique properties are

retrieved for the proposed technique to commence crossways of the DDoS assault after evaluating the time and collecting the details of established invitees for every state of activity.

4.7.1 Extraction of Features

The dissimilar processes of conveyance geometries to be appraised together with a broad range of standard packets inside the control organisation interval (CNPI), discrepancy of glide data sets within the cross - sections (VPI), and ordinary logistics within the choice period for thermal and irregular environments of transportation setups.

As indicated in Formula, CNPI is the sum of a wide range of float packages, or point per total flows, across the sampling interval (1). The DDoS physical attack is sending a huge number of plug-ins as a technique to decapitate the organiser, and ANPI is being used to expose the DDoS assault on the SDN open area.

$$CNPI = \frac{\sum_{i=1}^n \text{flow packet}_i}{\text{total flows}} \quad (1)$$

As shown in Formula, VPI is a trait of normal inconsistency of the quantity of current variation inference (2). The DDoS attempt on the SDN community was discovered using the VPI programme because most DDoS attackers build packets to transmit to hosts without thinking about the packed data packet and instead use void signals.

$$VPI = \sqrt{\frac{\sum_{i=1}^{\text{total flows}} (\text{flow packages}_i - CNPI)^2}{\text{total flows}}} \quad (2)$$

By analysing the RITI purpose of the SDN site visitors, which is the total of each and every time of the SDN traffic in accordance with a spread interval as shown in Formula (3), a suspicious network nature can be detected.

$$RITI = \frac{\text{Alltime duration of SDN traffic}}{\text{Selection interval}} \quad (3)$$

4.7.2 Concern Outcome Estimation

Facts with several dimensions have been resolved. Analyses and exercises The main challenge of multiclass in our research is DS. The second issue of a traditional algorithm's extended testing and training time has been overcome by using a linear kernel with an impact restriction of the business fault time period, 'C,' while considering the price of "gamma" and "OVS" proclamation function shapes. Our identification result is compared using the fake alarm rate, finding rate, and precision. The error margin of our classification stage is the false positive value, which is the wrong result on a typical performance.

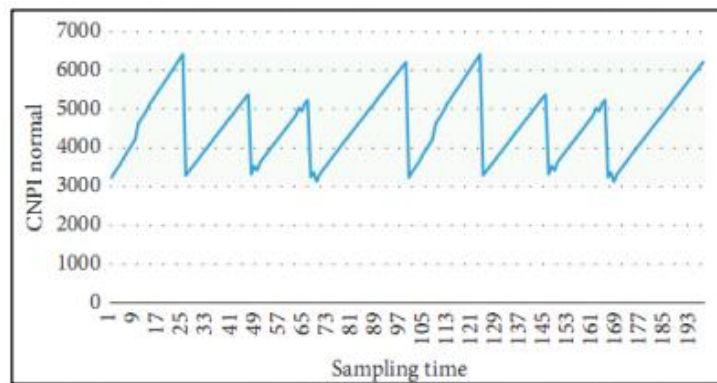


Figure 4.5. CNPI's Usual Traffic Flow Features

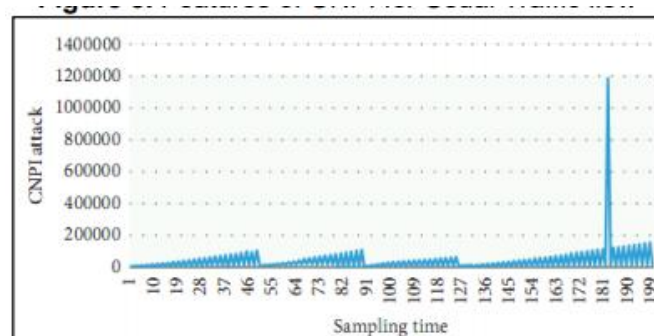


Figure 4.6. CNPI Features for Violent Traffic Flow

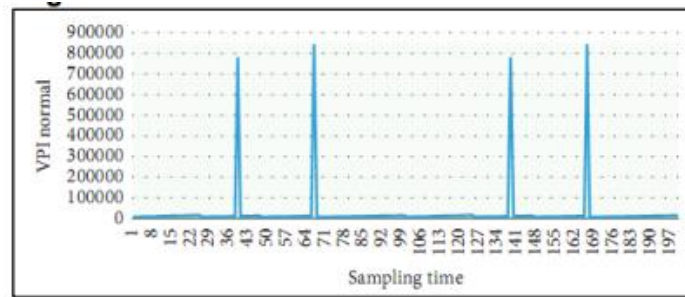


Figure 4.7. VPI topographies for public transportation

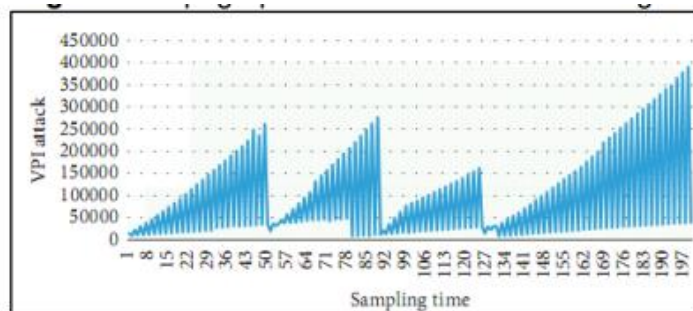


Figure 4.8. VPI Characteristics for Violence Carriages

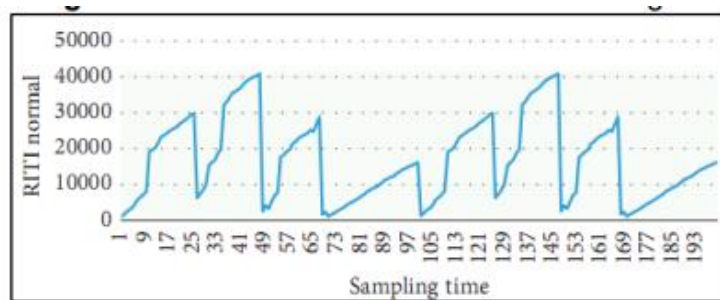


Figure 4.9. Features of RITI for Normal Traffic

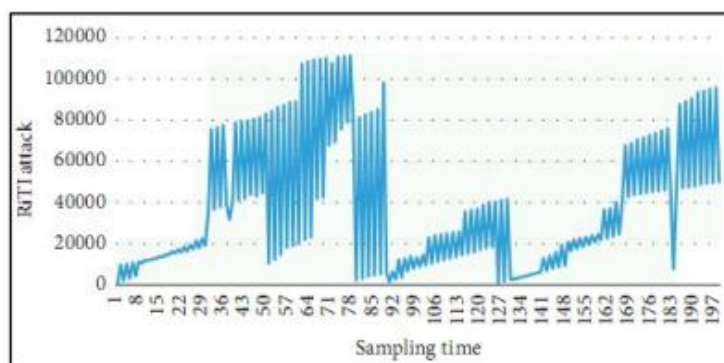


Figure 4.10. RITI's Attack Traffic Flow Features

The typical prediction accuracy is 0.89, the common fake alert rate is 0.23, and the mediocre recognition responsibility is 0.87, according to the experimental importance listed in table 4.1. Each service has a 45-second initiation period and a 60-second opting out period.

Table 4.1. Implications of the experiment

Split time	Exercise Data	Exciting Data	False alarm rate	Exposure Rate	Precision
0.1	90	10	0	1	1.0
0.2	80	20	0.06	0.92	0.92
0.3	70	30	0.02	0.98	0.97
0.4	60	40	0.03	0.97	0.97
0.5	50	50	0.01	0.99	0.99
0.6	40	60	0.01	0.98	0.97
0.7	30	70	0.01	0.99	0.99
0.8	20	80	0.02	0.96	0.96
0.9	10	90	0.03	0.97	0.97

4.8 SUMMARY

The Open Flow modifications are used to compose the SDN transportations. To construct the database, the spatial and imprecise parameters that initiate the SDN transportations are collected and retrieved. For training and requiring the ideal classification, a cross-validation method is used. Our proposed approach uses a linear kernel, and based on the results of the experiments, the suggested version's accuracy rate is 97 percent. On the SDN system, one of our destiny projects is a web monitoring device for DDoS aggression.

REFERENCES

1. Dang-Van T, Truong-Thu H. A multi-criteria based software defined networking system Architecture for DDoS-attack mitigation. REV Journal on Electronics and Communications. 2017 Oct 19;6(3-4).
2. Badotra S, Singh J. Open Daylight as a Controller for Software Defined Networking. International Journal of Advanced Research in Computer Science. 2017 May 15;8(5).

3. Gharvirian F, Bohlooli A. Neural network based protection of software defined network controller against distributed denial of service attacks. *International Journal of Engineering*. 2017 Nov 1;30(11):1714-22.
4. Kolahi SS, Treseangrat K, Sarrafpour B. Analysis of UDP DDoS flood cyber attack and defense mechanisms on Web Server with Linux Ubuntu 13. In 2015 International Conference on Communications, Signal Processing, and their Applications (ICCSPA'15) 2015 Feb 17 (pp. 1-5). IEEE.
5. Singh NA, Singh KJ, De T. Distributed denial of service attack detection using naive Bayes classifier through info gain feature selection. In Proceedings of the International Conference on Informatics and Analytics 2016 Aug 25 (pp. 1-9).
6. Benzekki K, El Fergougui A, Elbelrhiti Elalaoui A. Software-defined networking (SDN): a survey. *Security and communication networks*. 2016 Dec;9(18):5803-33.
7. Akamai A. Memcached Reflection Attacks: A NEW era for DDoS. Akamai Technologies, Cambridge, MA, USA. 2018.
8. Acharya S, Tiwari N. Survey of DDoS attacks based on TCP/IP protocol vulnerabilities. *IOSR Journal of Computer Engineering*. 2016 May;18(3):68-76.
9. Saleh Asadollahi D, Goswami B, Gonsai AM. Implementation of SDN using OpenDayLight Controller.
10. Tang F, Tiño P, Gutiérrez PA, Chen H. The benefits of modeling slack variables in svms. *Neural computation*. 2015 Mar 18;27(4):954-81.
11. Myint Oo M, Kamolphiwong S, Kamolphiwong T, Vasupongayya S. Advanced support vector machine-(ASVM-) based detection for distributed denial of service (DDoS) attack on software defined networking (SDN). *Journal of Computer Networks and Communications*. 2019 Mar 4;2019.

CHAPTER-5

BUILDING AN INSTRUCTION DETECTION SYSTEM USING MACHINE AND DEEP LEARNING APPROACHES IN VIRTUAL CLOUD SYSTEM

5.1 INTRODUCTION

Cloud Computing's enticing qualities continue to drive its acceptance and integration across a variety of industries, including industry, administration, academia, and media. However, enterprises face security risks like quality, reliability, and confidentiality when they upload critical data to public cloud storage providers. Furthermore, the cloud's open and distributed (decentralised) structure has made this type of computing vulnerable to cyber criminals and intruders. As a result, developing an anomalous network infiltration system that can identify and resist both internal and outside assaults in the cloud with detection accuracy efficiency and high false alerts is critical. Using a hybrid optimization framework (IGASAA) based on Improved Genetic Algorithm (IGA) and Simulated Annealing Algorithm, we offer an intelligent approach to automatically create an efficient and effective Deep Neural Network (DNN) based anomalous Network IDS (SAA). The IDS that resulted is known as “MLIDS” (Machine Learning based Intrusion Detection System). The Genetic Algorithm (GA) is enhanced by optimization tactics such as Parallel Processor and Fitness Function Hashing, which minimise completion time, convergence rate, and computing power consumption. Furthermore, SAA was integrated into IGA with the goal of improving its heuristic analysis. Our method entails employing IGASAA to find the best or near-best mixture of the most appropriate values of the parameters involved in the construction of a DNN-based IDS or affecting its performance, such as selecting features, data normalisation, DNN design, input vector, kernel size, and Momentum phrase, which ensures a high detection accuracy, good precision, and low detection limit. CloudSim 4.0 simulator system and 3 standard IDS dataset, including CICIDS2017, NSL-KDD version 2015, and CIDDS-001, were used for modeling and evaluation of the suggested technique.

5.2 BACKGROUND OF DNN

5.2.1 Operation of Deep Neural Network

(1) DNN maps input to intended or anticipated output by a series of layered transformations, with these stacked transformations acquired through interaction to training instances. DNN is a good example of how to learn. You offer DNN examples of what you want it to do, and it updates the network's parameters so that, once trained, it will provide the desired output for a unique input.

The weights of a layer, which are simply a group of numbers, define the modifications that it executes on its input. In other words, the weights of a layer control the modifications it performs. Learning can be described in this context as the process of determining the values of the connections of links of all nodes in the network in order to accurately map input instances to their corresponding objectives. A DNN contains hundreds of connections, and determining the optimal weight values for each is a difficult operation, especially when the number of one weight influences the value of another.

To train a network model, one must first determine how much the network's estimated output differs from the expected value. A regression model, also known as an objective function, is used to obtain this metric. The goal function computes the difference between the network's predicted output and the correct expected value for a given example. This is a metric for how well the network has learned a strange case. The goal of the training is to determine the weighting factors that will reduce the given error function. The obtained difference is then used as a feedback controller to alter the network's parameters in such a way that the loss value for the current example is minimised. The optimizer back - propagation method, which is the main algorithm of DNN, performs this correction.

Backpropagation process entails assigning numeric value to the weight matrices at the start, as seen in Fig. 5.1, so that the network simply conducts a series of random modifications. Initially, the output obtained by the network may be far from what it should have been, resulting in a very high loss score. This procedure is done until the weight values that minimise the loss

function are discovered. When the network's extracted features are as close as possible to the target values, the network is considered to have learned.

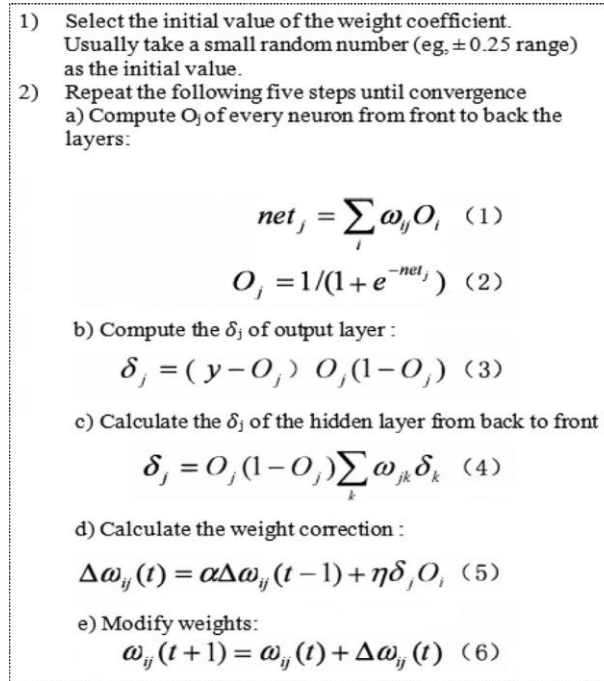


Fig. 5.1 – Operation of DNN learning process.

5.3 OPTIMIZATION STRATEGIES FOR GENETIC ALGORITHM

Because the fitness function is often the most processor intensive component of a GA, it makes sense to concentrate on improving it to get the best return on investment. In this section, we'll look at 2 optimization tactics that were employed in this research to improve GA performance by optimising the fitness function: Parallel Processing and Fitness Value Encryption.

5.3.1. Parallel processing

Improving the fitness function was one of the most straightforward ways to improve GA efficiency. The fitness function was usually the most computationally intensive component of GA, and it is frequently the bottleneck. As a result, it's an excellent candidate for multi-core optimization. It is feasible to estimate the fitness of several individuals concurrently utilising multiple processors, that makes a huge impact when there are hundreds of individuals to assess

each population. Java 8 comes with a number of handy packages that make distributed computation in our GA a lot easier. We can incorporate distributed computation in our fitness value using Java's IntStream without worrying about the intricate points of parallel processing (like the multiple cores we need to sustain); instead, it will generate an optimal no. of threads based on the number of cores accessible in our multi-core scheme. As a result, by utilising distributed computation, the fitness function would be able to run across different processes of the computer, allowing the GA to significantly decrease the amount of work it spends assessing individuals, thereby reducing the GA's overall time complexity and speeding up the convergence phase (2).

5.3.2. Fitness Value Hashing

Another technique for reducing the amount of time required computing fitness function is to use a hash table to store previously computed fitness values (3). Due to various changes and recombinations of people, solutions obtained previously will frequently be revisited throughout GA. As GA progresses and begins to find answers in a smaller portion of the search space, this periodic revisiting of problems becomes much more prevalent. The fitness value of a solution must be evaluated each time it is examined, spending processing capacity on redundant computations.

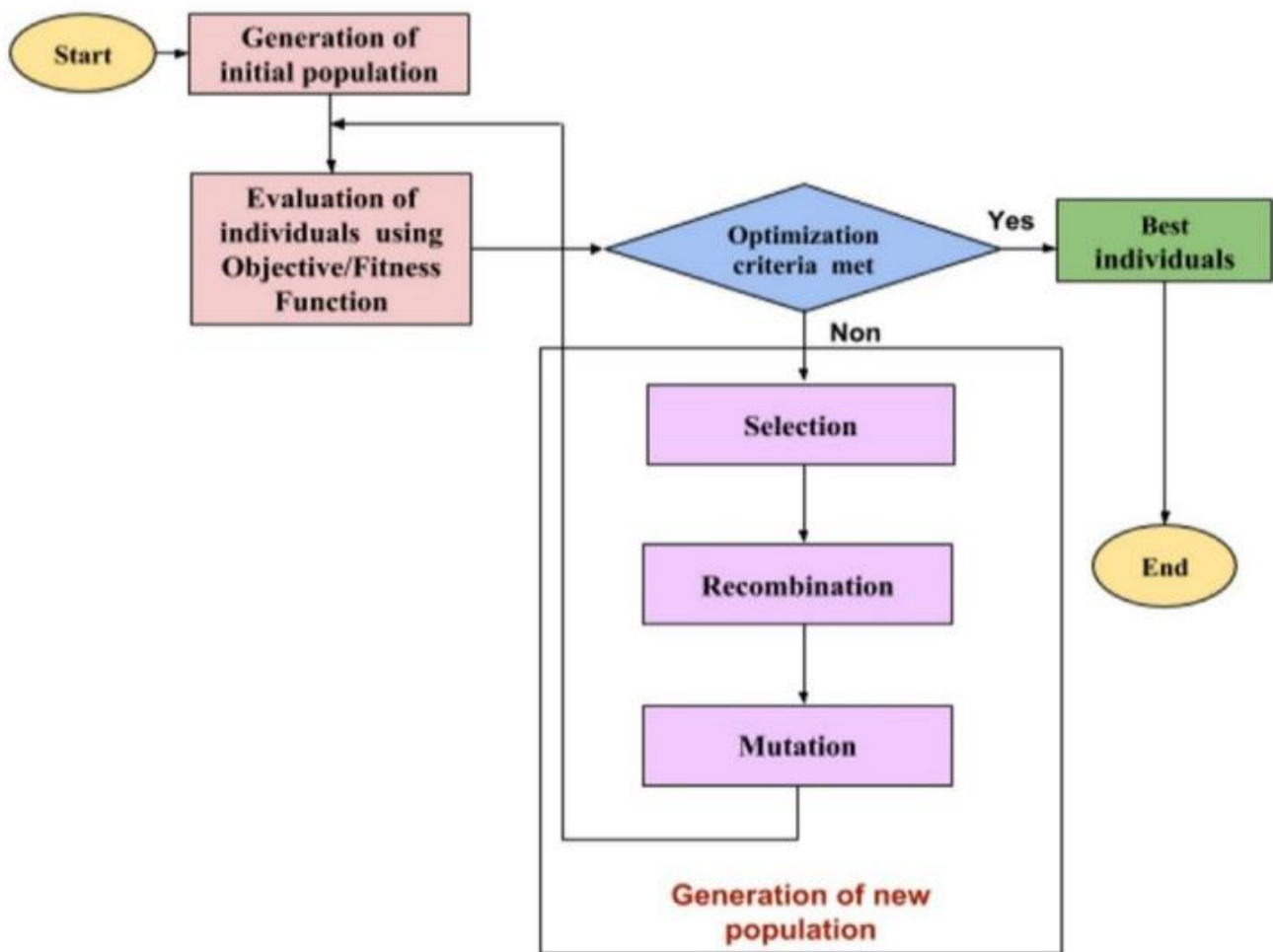


Fig. 5.2 – Flow of Genetic Algorithm

The procedure of a GA is depicted in Figure 5.2. The GA process starts with a set of initial remedies and then works towards changing solutions with better “goodness” as measured by the fitness value using a combination of techniques related to an evolution change. The fitness of these chromosomes is tested every generation. The fitness method is used to calculate the fitness of the chromosomes, and then the strongest chromosomes are chosen. The chromosomes with a low fitness value are thrown out. To establish a new population, the chosen fit chromosomes endure crossover and mutation. The next generation will be based on this new population. Usually, the algorithm ends when a certain number of generations have passed or a certain fitness value has been reached. Three operators make up a genetic algorithm. They are replication, crossover, and mixing, as well as mutation (4).

5.4 SIMULATED ANNEALING ALGORITHM

By analogy to probability theory, Accelerated Annealing is a meta-heuristic and a common search method that has shown to be useful in tackling many challenging problems, such as NP-hard computational optimization. The concept of SA was inspired by a paper written by (5). The algorithm used in this paper approximated material melting in a heat bath. In quantum theory, this is referred to as annealing. It is a physical thing that is frequently used to relax a system to a condition with the least amount of free energy. A solid in a heat bath was heated by gradually rising the bath's temp till the solid turns into liquid, and then the temperature is gradually reduced. In the liquid phase, all hard particles organize themselves at random. The particles are grouped in a properly organized lattice in the initial state, and the system's energy is low. Only if the maximum temp was reasonably high and the melting is sufficiently gradual can the solid reach its ground state. Otherwise, rather of freezing into the base state, the material will be frozen into a metastable form.

5.5 THE PROPOSED SYSTEM

This section explains our new suggested IDS in depth and provides the model for it.

5.6.1. Approach of our proposed system

In this paper, we offer a unique hybrid framework (IGASAA) that integrates an improved Genetic Algorithm (IGA) and Simulated Annealing Algorithm (SAA) to automatically design a network intrusion detection system (NIDS) based on Deep Neural Network (DNN) . Parallel Computing and Fitness Value Hashing are two optimization methodologies that improve the Genetic Algorithm (GA), reducing completion time, resolution time, and processing power (6). With one input layer, 2 hidden layers, and 1 output layer, our DNN is a Back - Propagation Network (BPNN). The number of nodes in the input layer correlates to the number of qualities in the network example vector collected from the IDS database and submitted to DNN, whereas the no. of nodes in each hidden units will be produced by IGA. The output unit consists of one node that returns a value of 1 if the input sequence is classified as regular traffic by DNN; otherwise, it returns a value of 0 to signify an intrusion. Our strategy is divided into four stages. We reviewed numerous studies related to IDS based on DNN and BPNN in two first levels.

The first phase was devoted to determining the most important factors used to build that sort of classifier or that have an effect on its performance. Table 5.1 shows that, at the conclusion of our research, the most essential parameters are (7):

Table 1 – List of parameters influencing the performance of a BPNN or a DNN based IDS and their different values.			
	Number of attributes/features	Normalization	Activation function
Different values	<ul style="list-style-type: none"> - 12 attributes NSL-KDD (Chiba et al., 2018; Lokeswari and Rao, 2016) - 10 attributes CIDDs-001 (Tama and Rhee, 2017) - 70 attributes CICIDS2017 (Ahmim et al., 2018) - 14 attributes Kyoto 2006+ (Song et al., 2011; Musbau and Alhassan, 2018) 	<ul style="list-style-type: none"> - Mean range [0,1] (Min-Max) (Wang et al., 2009) - Statistical normalization (Z-score) (Kumar and Yadav, 2014) 	<ul style="list-style-type: none"> - Hyperbolic - tangent (Sen et al., 2015) - Sigmoid (Gaidhane et al., 2014)

The number of selected attributes in the input layer that represents the number of nodes.

Data normalisation is the process of converting one set of data into another set of data.

The number of vertices in the secret layer of a neural network's architecture (s).

Activation or transfer function is a term used to describe a function that allows you to do something.

Momentum word. Learning rate.

The second stage is comparing the findings of the various studies in order to choose between two and four suitable values for each of the above-mentioned parameters that have produced the best intrusion detection results.

IGA will create the number of nodes in both hidden units of the DNN, as well as the values of Learning rate and Momentum term, at random in our research. IGA algorithm is capable to find the ideal values of certain parameters using genetic processes such as selection, elitism, crossover, and mutation. Some lack traffic heterogeneity and quantity, others don't cover a wide range of known threats, while still others are lacking or concealing elements found in the most used communication protocol. The CICIDS 2017 dataset (8), created by the Canadian Institute of Cybersecurity in 2017, resolves these concerns (9). Confidentiality, Attack Variety, Complete Capture, Complete Interface, Complete Network Architecture, Accessible Protocols, Whole Traffic, Image Set, Documentation, Stratification, and Classifying (10) are the eleven essential features of a good IDS dataset.

The third step is as follows: The representation of chromosomes and the Fitness Factor must both be properly specified for successful usage of IGA.

- ❖ **Chromosome embedding:** We used the binary format for chromosomes in our research. Each chromosome represents a potential combination of the previously described per-tinent variables that will be utilised to build an instance of IDS-based DNN. As shown in Table 5.2, each factor represents a gene on the chromosome. As a result, each chromosome is represented as a 58-bit binary string. Binary substrings relating to a chromosome's learning rate and movement term genes are transformed to decimal values, then normalised using the Min-Max normalisation technique to produce values between 0 and 1, which will be used as the Learning rate and Momentum term of the IDS generated predicated on that chromosome.

Table 2 – Structure of chromosome of IGA and some possible values of its genes.						
	Genes of chromosomes used by IGA					
	Normalization	Activation function	Nb of nodes in hidden layer 01	Nb of nodes in hidden layer 02	Learning rate	Momentum term
Number of bits to encode the gene	01	01	08	08	20	20
Possible/number of values	- 0 (Mean range [0,1] or (Min-Max) or - 1 (Statistical Normalization) or (Z-score)	- 0 (Hyperbolic tangent) - 1 (Sigmoid)	256	256 values	2^{20} values	2^{20} values

Fitness function or evaluation function: We have chosen the AUC metric (11) as a score (fitness function) of individuals of IGA to assess their adaptabil- ity to the optimization problem. The AUC is a performance metric of IDSs, that represents the ability to avoid misclas- sifications of network packets. From our point of view, it is a good trade-off between DR (Detection Rate) metric and FPR (False Positive Rate) metric. In effect, this is due to the fact that AUC is the arithmetic mean of DR and TNR (1-FPR) as shown by Eq. (1) of the AUC

$$AUC = \frac{(DR + TNR)}{2} = \frac{(DR + (1 - FPR))}{2}$$

As it is known, a good IDS is one that achieves a high detection rate (DR) and a low positive rate (FPR). As demonstrated by Eq. (1) , as the value of the DR measure increases and that of FPR measure decreases, consequently, the value of AUC increases. Therefore, from our point

of view, AUC is the best metric for evaluating an IDS. That is the reason of choice of AUC as fitness function.

The fourth stage : As shown by Fig. 5.3 , IGA process begins with a randomly generated population of 1000 individuals (potential solutions) represented by their chromosomes; each chromosome takes the form of a binary string of 58 bits. Then, this population evolves through several generations by means of genetic operations such elitism, selection, recombination (crossover) and mutation until stopping or optimization criterion of IGA is met. At each generation, for each chromosome, the Fitness Hash Table (FHT) is checked to verify if this chromosome is already visited, in this case, its fitness value is pulled from FHT.

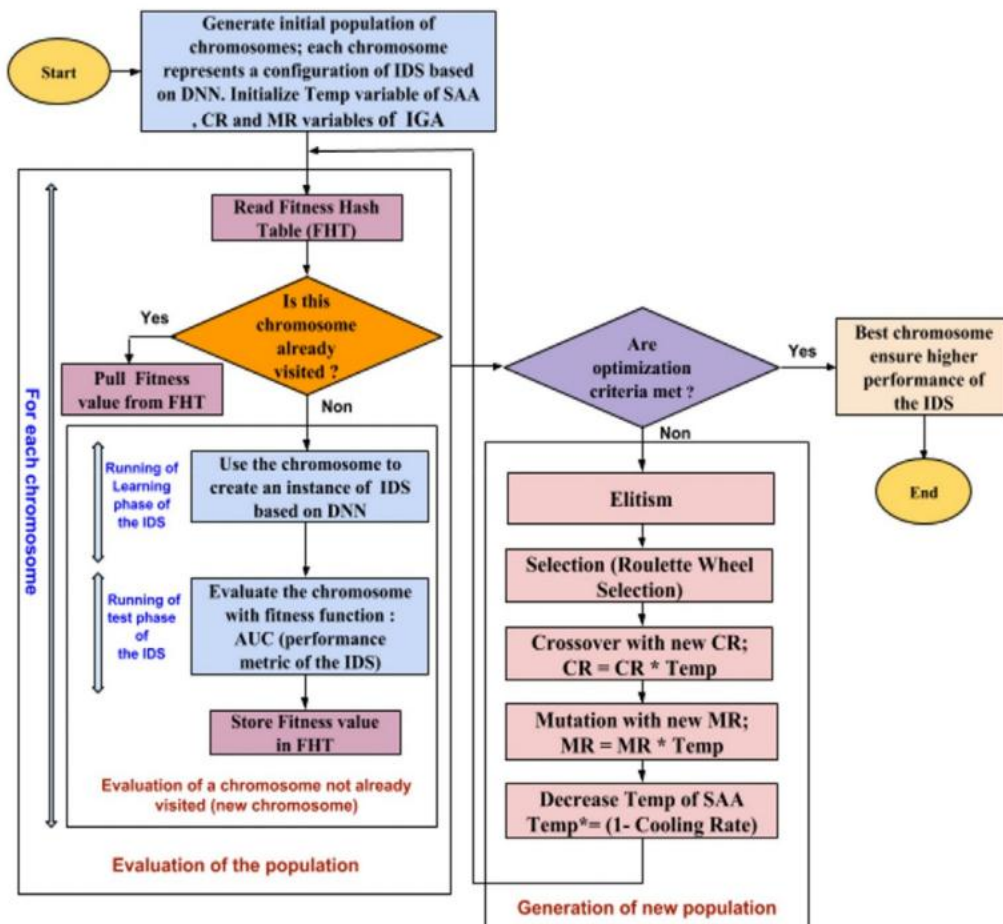


Fig. 5.3 – Workflow of proposed system

Otherwise, this chromosome is used to create an instance of an IDS based on DNN. Afterwards, this IDS firstly goes through the learning phase, then passes to the test/evaluation phase and returns the values of performance metrics calculated at the end of last phase. Among those performance metrics, we select the pertinent of them, namely AUC metric to serve as “Fitness Function” for evaluation of goodness of chromosomes, and the AUC (fitness value) is stored in FHT. From one generation to the next, IGA converges towards the global optimum through genetic operations cited previously. Finally, the best individual (chromosome) is picked out as the final result once the optimization criterion is met. In our work, termination condition adopted for IGA is production of 200 generations. Hence, the best chromosome resulted corresponds to the optimal or near-optimal values of parameters used to build an ideal IDS based DNN, which yields high detection rate and low false alarm rate.

In IGA module, we have employed the following algorithms/methods:

- Elitism.
- Roulette Wheel Selection.
- Single point Crossover.
- Bit flip mutation.

5.6. ROLE OF SIMULATED ANNEALING ALGORITHM IN THE PROPOSED SYSTEM

The aim of using Simulated Annealing Algorithm in our framework of optimization is to optimize Improved Genetic Algorithm (IGA) process. Simulated Annealing Algorithm is a hill climbing algorithm which initially accepts worse solutions at a high rate; then as the algorithm runs, it gradually reduces the rate in which worse solutions are accepted. One of the easiest methods to implement this characteristic into a genetic algorithm is by updating the mutation and crossover rate to start with a high rate then gradually decrease the rate of mutation and crossover as the algorithm progresses. This initial high mutation and crossover rate will drive IGA to search a large area of the search space. Then as the mutation and crossover rate is

slowly reduced, IGA should begin to focus its search on areas of the search space where fitness values are higher.

To vary the mutation and crossover rate/probability, we have used a temperature variable, which starts high, or “hot”, and slowly decreases, or “cools” by means of a *Cool rate* function as the algorithm runs. This heating and cooling technique is directly inspired by the process of annealing found in metallurgy. As displayed in Fig. 5.3 , at the end of each iteration/generation of IGA, the temperature is cooled slightly, which decreases the mutation and crossover rate that will be used in the next generation of IGA (11).

5.7. Framework of our proposed MLIDS

In this section, with the purpose of explanation of operation of our framework, we use CICIDS 2017 as IDS dataset. However, experimentation has been performed with three datasets, that is to say, CICIDS 2017, NSL-KDD and CIDDS-001 datasets.

Our system “MLIDS” passes firstly through an optimization stage by using IGASAA in order to find the optimal or near- optimal values of the parameters used to build an ideal IDS based DNN. Consequently, it becomes ready to operate in operation/normal mode. The framework of our system MLIDS in optimization mode consists of four modules as illustrated Fig.5. 4 .

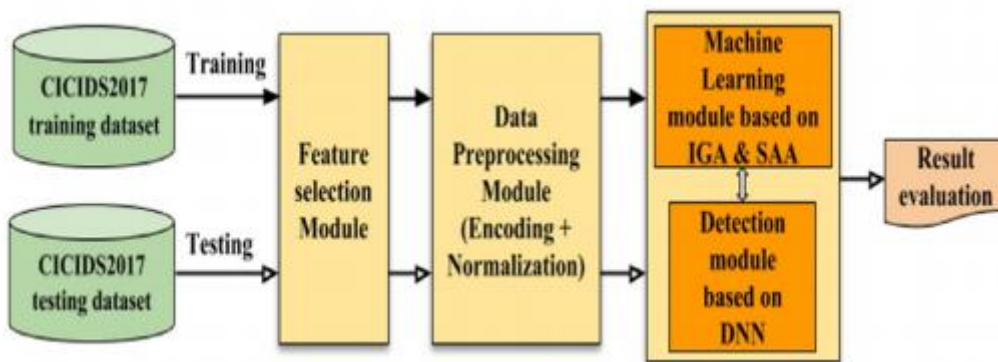


Fig. 5.4 – Framework of MLIDS in optimization stage.

❖ *Feature selection module*: Feature selection is the most critical stage in building intrusion detection models. Our intrusion detection model incorporates a feature selection module

mainly to select useful features for intrusion detection. This module allows selection of a set of 70 relevant features among 80 features of CICIDS2017 dataset .

❖ *Data preprocessing module:* Data Preprocessing includes two operations; data conversion (Categorical encoding) and Normalization. “Categorical encoding” refers to the process of assigning numeric values to nonnumeric features/attributes, so as to make the processing task much simpler, as numeric data can be easily handled upon. Whereas, “Normalization or Scaling” refers to the process of scaling the feature values to a small range that can help to obtain better detection results and avoid numerical difficulties during the calculation. Our data-preprocessing module uses Min-Max normalization and Statistical normalization methods.

5.8. POSITIONS OF THE PROPOSED SYSTEM IN A CLOUD NETWORK

The goal of the proposed MLIDS is to detect intruders and suspicious activities in and around the cloud computing environment by monitoring network traffic, while maintaining confidentiality, availability, integrity and performance of cloud resources and offered services. It allows detecting and stopping attacks in real time impairing the security of the Cloud Datacenter. As shown in Fig. 5.5, we propose to deploy our NIDS on two strategic positions:

❖ *Front-end of cloud:* Placing NIDS on front end of Cloud helps to detect network intrusions or attacks coming from external network of Cloud, launched from zombie hosts or by hackers connected to the Internet who attempt to bypass the firewall in order to access the internal cloud, which can be a private one. Therefore, NIDS plays the role of the second line of defense behind the firewall to overcome its limitations (12,13), and acts as an additional preventive layer of security (13)

❖ *Back-end of cloud:* Positioning NIDS sensors on processing servers located at back end of Cloud helps to detect intrusions occurring on its internal network. In a virtual environment, we have many virtual machines on the same physical server, and they can inter-communicate through the virtual switch without leaving the physical server. Thus, network security devices on the LAN cannot monitor this network traffic; if the traffic does not need to pass through security appliances primarily a firewall, therefore, a loophole for all kinds of security attacks

will be opened. Thus, the starting point of an attacker/hacker is compromising only one VM, and using it as a springboard to take control of the other VMs within the same hypervisor. This is generally done without being monitored or detected, giving the attacker a huge hack domain. Moreover, the virtual environment is exposed to various threats and risks, centered mostly on the hypervisor: Hyper jacking, VM escape, VM migration, VM theft and Inter-VM traffic. Our NIDS is designed to monitor that virtual traffic, and also the flow of traffic from or to the processing server on the physical network. We haven't chosen to install the NIDS on each virtual machine because it will be an additional burden; it will weigh down the work of the VM. Further, such configuration requires multiple instances of NIDS, which makes complex management of NIDS whereas VMs are dynamically migrated, provisioned or de-provisioned.

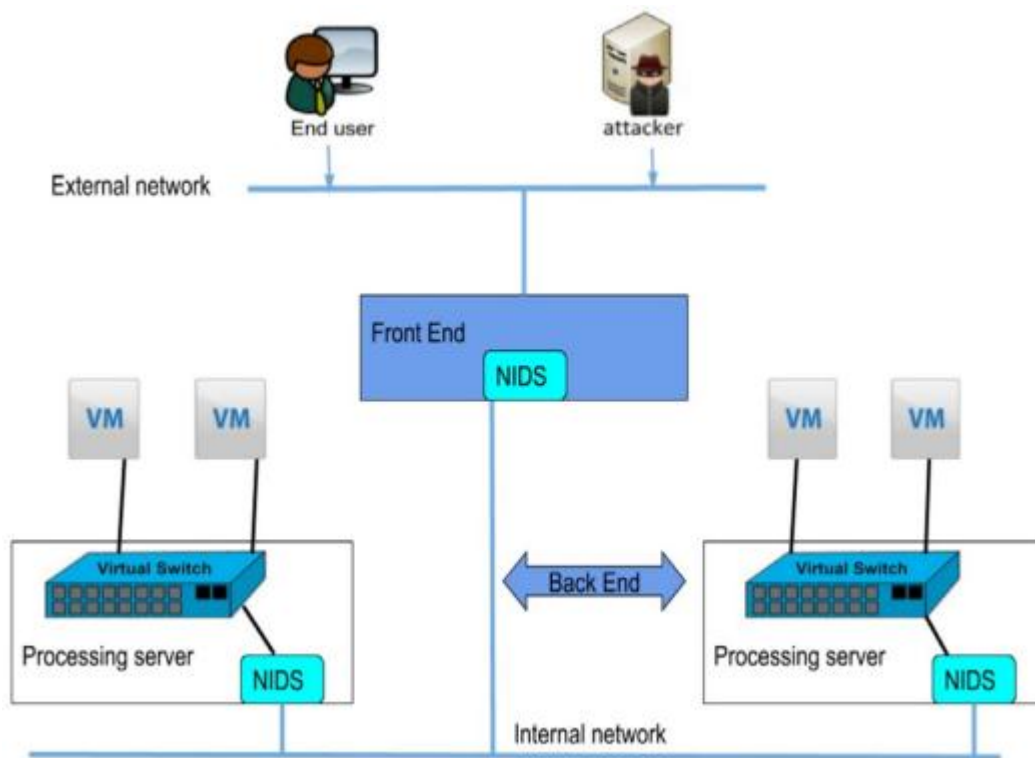


Fig. 5.5 – Positions of proposed MLIDS in a cloud network.

5.9. EXPERIMENTATION

5.9.1 Data preprocessing

With the aim to make the records in both training and test- ing subsets extracted from the three IDS datasets used in this study ready for processing by our proposed IDS, they must be handled using the two following operations:

- 1) *Numericalization or categorical encoding*: This operation refers to the process of assigning numeric values to nonnumeric features/attributes so as to make the processing task much simpler, as numeric data can be easily handled upon (14).
- 2) *Normalization*: The attributes with high values can domi- nate the results than the attributes with lower values. This dominance can be reduced by the process of normaliza- tion, i.e., scaling the values within certain range. Normal- ization is defined as is the process of enclosing the values of attributes to a specific range to minimize the complex- ity involved in handling data spread over an absolute range and type of values. Various features have values spread over large ranges and types. Hence, they are to be mini- mized to a specific range being useful to enable proper pro- cessing and analysis of the data (15). To normalize data, the mean-range [0, 1] (Min-Max) nor- malization and Statistical normalization (Z-score) meth- ods are used. The reason for choosing these approaches is that they yield better results in terms of time and classifi- cation rate (16).

Mean range [0,1] (Min - Max normalization): As shown by Eq. (2) , the mean range technique normalizes an attribute value by subtracting minimum value of that attribute from the current value. This value is further divided by the difference between maximum and minimum values of that attribute.

$$X' = \frac{x - \text{MinA}}{\text{MaxA} - \text{MinA}}$$

x and x' are value to be normalized and the normalized attribute value, respectively. MinA and MaxA are the minimum and maximum possible values for attribute A before normalization.

Statistical normalization (Z-score normalization):

$$X' = \frac{x - \mu}{\alpha}$$

The value x of an attribute A is transformed in x' according to formula (3). μ is the mean and α is standard deviation of given attribute

5.9.2 Machine learning optimization framework IGASAA

Table 5.4 presents the parameters of our machine-learning framework used in this study with the purpose to build automatically an optimal or near optimal IDS based on DNN.

Table 4 – Parameters of our machine learning framework based IGASAA.		
Component of the framework IGASAA	Parameters	Value
Improved Genetic Algorithm (IGA)	Length of chromosomes	58 bits
	Elitism number: the number of best chromosomes which will be copied without changes to a new population (next generation)	100
	Population size	1000
	Maximum number of generations	200
	Initial Crossover rate (Dynamic parameter)	0.95
	Initial Mutation rate (Dynamic parameter)	0.1
	Size of Fitness Hash Table (FHT)	10,000
Simulated Annealing Algorithm (SAA)	Initial Temperature (Dynamic parameter)	1.0
	Cooling rate	0.001

This framework combines an Improved Genetic Algorithm and Simulated Annealing Algorithm, and it is employed later during the experiments carried out with the CICIDS2017, NSL-KDD and CIDDs-001 datasets.

5.9.3. Experimentation based on CICIDS 2017 dataset

5.9.3.1. Description of CICIDS2017 dataset

To successfully build efficient anomaly detection and threat classification model, a big amount of data is required to train and test its detection accuracy. At the same time, most of the existing network traffic datasets that are publicly available are outdated, unreliable or unlabeled. Some of these suffer from lack of traffic diversity and volume, some do not cover the variety of known attacks, while others are missing or hiding features that are present in the most common network protocols (16). The CICIDS 2017 dataset (17) generated in 2017 by the Canadian Institute of Cybersecurity overcomes these issues. It represents a data set that satisfies the eleven indispensable characteristics of a valid IDS dataset, namely Anonymity, Attack Diversity, Complete Capture, Complete Interaction, Complete Network Configuration, Available Protocols, Complete Traffic, Feature Set, Metadata, Heterogeneity, and Labeling (18). This dataset contains benign traffic along with the most up-to-date common attacks, approaching real-world data as much as possible.

5.9.3.2 Data pre-processing

As outlined in the paper (19), CICIDS2017 dataset contains the following few shortcomings that must be handled:

- a) *Scattered presence*: It can be seen from Table 5.5 that CICIDS2017 dataset contains attack and benign information as five days traffic data. Thursday working hour afternoon and Friday data are well appropriate for binary classification. Likewise, Tuesday, Wednesday and Thursday morning data are best for designing multiclass detection model. Nevertheless, it should be noted that a best detection model should be able to detect attacks of any kind. Thus, to design such a typical IDS, the traffic data of all the day should be merged together to form a single dataset to be used by IDS (20). Thereby, we have merged the 8 files in one same file that comprises all benign and attacks rows. As result, the obtained dataset comprises 2,830,743 rows and 15 class labels (1 normal + 14 attack labels).
- b) *Irrelevant features*: There are some rows in dataset, which have the feature “Flow Bytes/s” equal to ‘Infinity’ or ‘NaN’. As the classifier based on DNN cannot utilize this feature either

in training or in testing phases due to its values, to resolve this concern, we have removed all those rows.

Table 5 – Description of files containing CICIDS2017 dataset.						
File	Name of Files	Description	Normal flows	Attack flows	Total	Class labels found
1	Monday-WorkingHours.pcap_ISCX.csv	Normal traffic captured on Monday, July 3, 2017	529,918	0	529,918	Benign (Normal human activities)
2	Tuesday-WorkingHours.pcap_ISCX.csv	Brute force attack on FTP and SSH servers captured on Tuesday, July 4, 2017	432,074	13,835	445,909	Benign, FTP-Patator, SSH-Patator
3	Wednesday-WorkingHours.pcap_ISCX.csv	DoS/DDoS and heartbleed attacks captured on Wednesday, July 5, 2017	440,031	252,672	692,703	Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed
4	Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv	Web attacks (brute force, XSS and SQL injection) captured on the morning of Thursday, July 6, 2017	168,186	290,782	458,968	Benign, Web attack-Brute Force, Web Attack-Sql Injection, Web Attack-XSS
5	Thursday-WorkingHours-Afternoon-Infiltration.pcap_ISCX.csv	Infiltration attacks captured on the afternoon of Thursday, July 6, 2017	288,566	36	288,602	Benign, Infiltration
6	Friday-WorkingHours-Morning.pcap_ISCX.csv	Botnet traffic captured on the morning of Friday, July 7, 2017	189,067	1966	191,033	Benign, Bot
7	Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	DDoS traffic captured on the afternoon of Friday, July 7, 2017	183,910	41,835	225,745	Benign, PortScan
8	Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	Port scan traffic captured on the afternoon of Friday, July 7, 2017	127,537	158,930	286,467	Benign, DDoS

5.9.4. Experimental results

From the complete CICIDS2017 dataset, we have extracted two independent subsets, namely training dataset and testing dataset, following the approach explained in the following section. Table 5.6 summarizes the distribution and size of these subsets.

Table 5.6 – Distribution and size of training and testing datasets.

Dataset	Attack records	Normal records	Total
Training dataset	20,000	20,000	40,000
Testing dataset	20,000	20,000	40,000

The experiments conducted on our proposed system show that at the end of IGA process optimized by SAA that is to say after 200 generations, the best individual (chromosome) found contains the genes displayed in Table 5.7 with its values. This fittest chromosome allows building the best machine learning IDS.

Table 5.7 – Genes of fittest chromosome found by IGASAA

Gene of chromosome	Value
Normalization	0 (Min-Max normalization)
Activation function	1 (Sigmoid Function)
Number of nodes in hidden layer 01	53
Number of nodes in hidden layer 02	27
Learning rate	8.412874497406361E-7
Momentum term	1.264876945375064E-4

Table 5.8 indicates configuration and also performances reached by that best MLIDS 1 based DNN using CICIDS2017 dataset, called “MLIDS_ CICIDS2017”.

Table 5.8– Configuration and performances of best MLIDS_ CICIDS2017

Parameters	Value	Performance Metric	Value
Number of nodes in Input layer	70	Accuracy	99.93%
Number of nodes in Hidden layer 01	53	Precision	99.95%
Number of nodes in Hidden layer 02	27	Detection Rate (DR)	99.92%
Number of nodes in Output layer	1	False Negative Rate (FNR)	0.08%
Activation function	Sigmoid	False Positive Rate (FPR)	0.05%
Normalization of data	Min-Max	True Negative Rate (TNR)	99.95%
Learning rate	8.412874497406361E-7	F-score	0.99
Momentum term	1.264876945375064E-4	AUC (Ability to avoid misclassifications)	99.93%

5.10. EXPERIMENTATION BASED ON NSL-KDD DATASET

5.10.1. Description of NSL-KDD dataset

Traditionally, most of the research work conducted on the field of intrusion detection utilized the popular benchmark dataset KDD99, which has a huge size and includes many duplicate and redundant records. The large size of the data set causes the classification task to be long and exhausting. Consequently, the researchers in general select a small portion of the training and testing data set for their experiments. The very high number of redundant records makes the classification task be biased toward the frequent samples. The detection for smaller classes i.e. U2R and R2L, which are more harmful attacks, will not be efficient enough.

Moreover, the existence of these redundant records in the test set will also cause the evaluation results to be biased by the algorithms, which have better detection rates on frequent samples. In this study, we used the NSL-KDD data set (21), which is the refined version of KDDcup99 and has eliminated some of the drawbacks of that dataset. It has the following advantages (21):

There are no redundant records in the training set, so the bias toward more frequent records will not happen during the learning process.

There are no duplicate records in the new test set, so the evaluation of the learners will not be biased by the methods which have higher detection accuracy for the frequent records.

The number of selected records from each difficulty level group is inversely proportional to the percentage of records in the original KDD data set. As a result, the classification rates of distinct machine learning methods vary in a wider range, which makes it more efficient to have an accurate evaluation of different learning techniques.

5.10.2 Experimental results

For experimentation, we have utilized two types of NSL-KDD dataset version 2015. The first one is NSL-KDD Train+ (training dataset) and the second is NSL-KDD Test+ (testing dataset) (Section 6.5.1). For feature selection, we have opted for A modified Kolmogorov-Smirnov Correlation Based Filter Algorithm, which allows selection of a set of 12 relevant features among 41 features of NSL-KDD datasets. The set of 12 features resulted

has given good results in anomaly-based intrusion detection (22,23). The experiments driven on our model show that at the end of running of our framework IGASAA, that is to say after 200 generations, the best individual (chromosome) found leads to construction of the best MLIDS 2, called “MLIDS_ NSL-KDD”. Table 5.9 displays configuration and performances attained by that IDS.

Table 5.9 – Configuration and performances of best MLIDS_ NSL-KDD

Parameters of configuration	Value	Performance metric	Value
Number of nodes in Input layer	12	Accuracy	99.86%
Number of nodes in Hidden layer 01	8	Precision	99.93%
Number of nodes in Hidden layer 02	5	Detection Rate (DR)	99.83%
Number of nodes in Output layer	1	False Negative Rate (FNR)	0.17%
Activation function	Sigmoid	False Positive Rate (FPR)	0.09%
Normalization of data	Min-Max	True Negative Rate (TNR)	99.91%
Learning rate	8.412874497406361E-7	F-score	0.99
Momentum term	1.264876945375064E-4	AUC (Ability to avoid misclassifications)	99.87%

5.10.3 Experimentation based on CIDDs-001 dataset

A. Description of CIDDs-001 dataset

CIDDS-001 (Coburg Network Intrusion Detection Dataset) (CIDDS, 2019) is a labeled flow based dataset created by (24) in a Cloud environment based on OpenStack platform. This dataset contains unidirectional NetFlow data. It consists of traffic data from two server's i.e. OpenStack and External server. The dataset is generated by emulating small business environment which consist of OpenStack environment having internal servers (web, file, backup and mail) and an External Server (file synchronization and web server) which is deployed on the internet to capture real and up-to-date traffic from the internet. It consists of three logs files (attack logs, client configurations and clientlogs) and traffic data from two servers where each server traffic comprises of 4 four week captured traffic data (25).

5.10.3.1 Experimental results

The Original version of CIDDS-001 contains 5 classes, i.e. normal, suspicious, unknown, attacker, and victim, and since our objective is to evaluate anomaly based IDS, we only included normal and attacker classes in our experimental dataset. Thus, as shown by Table 5.10, the reduced version of CIDDS-001 dataset used in this work contains 953,298 normal instances and 65,652 attacker instances, extracted from both week 1 and week 2 of OpenStack and ExternalServer traffic folders respectively. This reduced version of CIDDS-001 dataset is then splitted into train and test subsets using a configuration of 60% for training and 40% for testing. Table 5.11 presents distribution and size of those subsets.

Table 5.10 – Construction of a reduced version of CIDDS-001 dataset

Traffic folder	File source	Normal records	Attacker records
OpenStack	CIDDS-001-internal-week1.csv	924,862	63,263
ExternalServer	CIDDS-001-external-week2.csv	28,436	2389
reduced CIDDS-001 dataset		953,298	65,652

Table 5.11 – Distribution and size of testing and training CIDDS-001 subsets.

Datasets	Normal records	Intrusive records	Total records
Training subset	571,979	39,392	611,371
Testing subset	381,319	26,260	407,579

The experiments conducted on our model points that on completion of executing of our framework IGASAA, that is to say after 200 generations, the best individual (chromosome) found allows building the best MLIDS 3, called “MLIDS_CIDDS001”

5.11. Conclusions and future work

In order to develop an efficient and an effective anomaly network intrusion system (ANIDS) for detection and prevention of both inside and outside assaults in cloud environment with high detection precision and low false warnings, we have adopted an intelligent approach to build automatically such IDS based on Deep Neural network (DNN). Our method consists of using a hybrid framework called “IGASAA” that combines machine learning techniques, namely Improved Genetic Algorithm (IGA) and Simulated Annealing Algorithm (SAA), with the purpose of searching the optimal values of the parameters included in construction of IDS based DNN (IDSDNN) or affecting its performance. GA was improved through optimization strategies that are Parallel Processing and Fitness Value Hashing. In addition, SAA was incorporated to IGA with the aim to optimize its heuristic search. In IGA process, we have used binary encoding for chromosomes, while AUC metric was selected as a fitness function for evaluating the goodness of the chromosomes generated versus the optimization problem in hand. In each generation of IGA, each chromosome produced is used to create an instance of IDSDNN, which thereafter goes through learning phase and a test phase. At the end of the last phase, AUC measure is computed. IGA process begins with a randomly generated population, which evolves through elitism, selection, recombination (crossover) and mutation. Finally, the best individual (chromosome) is picked out as the final result once the optimization criterion is met. In our work, termination condition adopted for IGA is production of 200 generations. As result, at the end IGA process, the optimal or near-optimal values of parameters used to build an ideal IDSDNN are found, which allows constructing a powerful machine learning IDS called “MLIDS” reaching high detection rate and low false positive rate.

REFERENCES

1. Peng J, Aved AJ. Information Preserving Discriminant Projections. InICAART (2) 2020 Jan 1 (pp. 162-171).

2. Ahmim A, Maglaras L, Ferrag MA, Derdour M, Janicke H. A novel hierarchical intrusion detection system based on decision tree and rules-based models. In 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS) 2019 May 29 (pp. 228-233). IEEE.
3. Aminanto ME, Kim H, Kim KM, Kim K. Another fuzzy anomaly detection system based on ant clustering algorithm. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. 2017 Jan 1;100(1):176-83.
4. Amini M, Rezaeenour J, Hadavandi E. A neural network ensemble classifier for effective intrusion detection using fuzzy clustering and radial basis function networks. International Journal on Artificial Intelligence Tools. 2016 Apr 22;25(02):1550033.
5. Woolf N. DDoS attack that disrupted internet was largest of its kind in history, experts say. The Guardian. 2016 Oct 26;26.
6. Bansal A, Kaur S. Extreme gradient boosting based tuning for classification in intrusion detection systems. In International conference on advances in computing and data sciences 2018 Apr 20 (pp. 372-380). Springer, Singapore.
7. Borah S, Panigrahi R, Chakraborty A. An enhanced intrusion detection system based on clustering. In Progress in Advanced Computing and Intelligent Engineering 2018 (pp. 37-45). Springer, Singapore.
8. Chiba Z, Abghour N, Moussaid K, Rida M. A cooperative and hybrid network intrusion detection framework in cloud computing based on snort and optimized back propagation neural network. Procedia Computer Science. 2016 Jan 1;83:1200-6.
9. Deshpande PS, Sharma SC, Peddoju SK. A network-based intrusion detection system. In Security and Data Storage Aspect in Cloud Computing 2019 (pp. 35-48). Springer, Singapore.
10. Ghanshala KK, Mishra P, Joshi RC, Sharma S. BNID: a behavior-based network intrusion detection at network-layer in cloud environment. In 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC) 2018 Dec 15 (pp. 100-105). IEEE.
11. Ghosh P, Jha S, Dutta R, Phadikar S. Intrusion detection system based on BCS-GA in cloud environment. In International Conference on Emerging Research in Computing,

- Information, Communication and Applications 2016 Jul 29 (pp. 393-403). Springer, Singapore.
12. Hatef MA, Shaker V, Jabbarpour MR, Jung J, Zarrabi H. HIDCC: A hybrid intrusion detection approach in cloud computing. *Concurrency and Computation: Practice and Experience*. 2018 Feb 10;30(3):e4171.
 13. Idhammad M, Afdel K, Belouch M. Distributed intrusion detection system for cloud environments based on data mining techniques. *Procedia Computer Science*. 2018 Jan 1;127:35-41.
 14. Kang MJ, Kang JW. A novel intrusion detection method using deep neural network for in-vehicle network security. In *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)* 2016 May 15 (pp. 1-5). IEEE.
 15. Lokeswari N, Rao BC. Artificial neural network classifier for intrusion detection system in computer network. In *Proceedings of the Second International Conference on Computer and Communication Technologies 2016* (pp. 581-591). Springer, New Delhi.
 16. Ma T, Yu Y, Wang F, Zhang Q, Chen X. A hybrid methodologies for intrusion detection based deep neural network with support vector machine and clustering technique. In *International Conference on Frontier Computing 2016* Jul 13 (pp. 123-134). Springer, Singapore.
 17. Mohammadi S, Amiri F. An efficient hybrid self-learning intrusion detection system based on neural networks. *International Journal of Computational Intelligence and Applications*. 2019 Mar 27;18(01):1950001.
 18. Pajouh HH, Dastghaibiyfard G, Hashemi S. Two-tier network anomaly detection model: a machine learning approach. *Journal of Intelligent Information Systems*. 2017 Feb 1;48(1):61-74.
 19. Roy SS, Mallik A, Gulati R, Obaidat MS, Krishna PV. A deep learning based artificial neural network approach for intrusion detection. In *International Conference on Mathematics and Computing 2017* Jan 17 (pp. 44-53). Springer, Singapore.
 20. Sharafaldin I, Gharib A, Lashkari AH, Ghorbani AA. Towards a reliable intrusion detection benchmark dataset. *Software Networking*. 2018 Jan 31;2018(1):177-200.

21. Sharma R, Chaurasia S. An enhanced approach to fuzzy C-means clustering for anomaly detection. In Proceedings of first international conference on smart system, innovations and computing 2018 (pp. 623-636). Springer, Singapore.
22. Singh DA, Priyadharshini R, Leavline EJ. Cuckoo optimisation based intrusion detection system for cloud computing. International Journal of Computer Network and Information Security. 2018 Nov 1;11(11):42.
23. Verma P, Anwar S, Khan S, Mane SB. Network intrusion detection using clustering and gradient boosting. In 2018 9th International conference on computing, communication and networking technologies (ICCCNT) 2018 Jul 10 (pp. 1-7). IEEE.
24. Song Q, Guo Y, Shepperd M. A comprehensive investigation of the role of imbalanced learning for software defect prediction. IEEE Transactions on Software Engineering. 2018 May 15;45(12):1253-69.
25. Panigrahi R, Borah S. A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. International Journal of Engineering & Technology. 2018 Dec;7(3.24):479-82.

CHAPTER-6

CASE STUDY OF REAL TIME APPLICATION SET-UP IN CLOUD COMPUTING

6.1 INTRODUCTION

This chapter examines a variety of case studies that are relevant to real-world applications, such as KVM, Xen, and the advent of green computing in the cloud, which were explored in Chapter 2. Lastly, this chapter focuses on a single case study that is particularly valuable for statistical analysis in remote settings. There are many methods for either transactional or spatial databases that have been proposed to trim the frequent patterns and classification methods: here, a system is developed to find proper spatial association rules, which is solely portrayed in GIS data models and spatial by one-to-one and one-to-many connections. This chapter explains how to enhance spatial mining algorithm with an algorithm. There are two important steps in the proposed algorithms: The first step is to automate the GIS module's geographic data preparation operations. Second, all well-known GIS connections that compute the link between several attributes are discarded. GIS, information extraction, distributed data, statistical analysis, and green computing are some of the terms used in this paper.

6.1.11 Kernel-Based Virtual Machine

A hypervisor embedded within the Linux kernel is known as a KVM. In purpose, it's comparable to Xen, but it's a lot easier to set up. To use the hypervisor, simply load the relevant KVM kernel packages, and the hypervisor will be up and running. To use KVM, you'll need a processor that implements Intel's VT-x extensions or AMD's AMD-V extensions, much like with Xen's full virtualization. KVM is a complete Linux virtualization system. It is centered on additions to CPU virtualization . KVM was a new Linux system that uses these modifications to provide a virtual machine monitor ability to Linux . KVM allows you to construct and run several virtual machines that seem as regular Linux processes and are fully integrated into the system. It's based on the x86 design and enables hardware virtualization methods like Intel's VT-x and AMD's AMD-D.

6.1.12 Xen

Xen [2] was an accessible type-1 or bare-metal virtualization that allows several instances of a linux kernel, or even multiple software platforms, to run concurrently on a single device (or host). Xen was the only active type-1 hypervisor on the market. Virtualization technology, IaaS, consumer virtualization, privacy software, integrated and physical appliances are just a few of the commercial and free source products based on Xen. Users can utilise Xen to boost server utilisation, combine server farms, simplify their operations, and lower their ownership costs.

6.1.3 Secure Data Analysis in GIS

According to a cloud storage poll, this is the Internet age, and each user wants to save and recover their public and private data. When data is saved on the server, the issue arises when the user wants to access it, because there are a variety of approaches available in the data mining field, such as association rule, categorization, grouping, and so on. There are 2 main strategies as well: the first is forecasting [3], in which the database administrator forecasts the link between end consumers or a set of attributes. The second type is descriptive, in which the database administrator explains the user's valuable information. Association rule methods are highly important in data mining algorithms for determining the relationship between a large number of databases. Clustering was the 2nd strategy, in which qualities are deleted or grouped based on their values. The final strategy is classification, which classifies attributes based on user factors such as age, education, and so on.

6.1.4 Database

A database was a set of data, with data representing useful data acquired from a physical entity. The database control system was the system that manages the acquired information. This system is essential for the company, business, and so on. Consider a campus database that contains information about professors, employees, students, coursework, divisions, and other items that is updated on a regular basis. In the network, there are several sorts of database servers, such as centralised and distributed databases. The dispersed database approach is faster than the centralised data structure, but it requires more work in terms of privacy.

6.1.5 Data Mining and Techniques

Data mining was the process of extracting usable information or patterns from large databases, like data warehouses. The data warehouse was a multifunctional repository in which new data is added but no alteration of existing data is permitted. The KDD [6] includes a process called data mining.

6.1.6 Distributed Database

A distributed system is one in which information is actually stored on multiple computers but is linked via a managed network. The distributed system is the fastest and least memory-intensive way of data transmission, and it is also the most expensive because privacy and additional administrative chores, like replication and duplication, are necessary.

6.1.7. Spatial Data Mining

The way items link in space around the planet is referred to as spatial characterisation. Spatial data is quantified data that includes an object's length, height, breadth, and other attributes. A spatial database is a collection of this sort of listed database table that defines a global geographic structure. This will be expressed by their graphical viewpoints, which will be a connection of their pixel-position in the three-dimensional architecture. A spatial database is a collection that has been enhanced to organize and maintain geometric area. Coordinates, lines, vectors, and polygons are common elements in this form of data. More complicated data, such as three-dimensional objects, hierarchical coverage, and longitudinal networks, can be handled by some geographical databases. The use of data mining to geographic models is known as spatial data mining.

6.1.8. Secure Multi-Party

Computation SMC is based on the notion that all parties wishing to communicate with one another are either untrustworthy of one another or distrustful of the channels of communication. They still wish to compute some basic operations while maintaining the security of their relevant information. A concrete conceptual foundation for security is provided by the skeleton of safe multi-party processing.

TTP (Trusted Third-Party) Model: The TTP model assumes that the data cannot be inferred from anybody else. The secure protocol's main goal is to achieve that degree of security. When data is disseminated in a distributed environment, the TTP model works because each system owner has their own private data and no one needs to gain their personal data with other database owners. As a result, one of them is chosen as the authorized third party in charge of estimating or maintaining all of the secure and private information from all of the other cloud providers in the environment.

6.1.9 Association Rule Mining Problem

Researchers have discovered that ARM is one of the basic processes of data analysis in the previous decade. ARM was the most essential data mining procedure for discovering all of the relationships between common patterns, and it doesn't require any supervision. ARM analyses data of varying lengths to produce understandable results. The structure of modern organisations is regionally spread. Every place, on the other hand, saves its ever-increasing amount of daily data locally. Because of the high interprocess communication costs generated in such organised data, centralised data mining is unable to uncover feasible meaningful patterns. Remote data mining is used to solve this problem. Let $I = I_1, I_2, I_m$ be a collection of m distinct characteristics, T be a transaction containing a set of objects that are a subset of I , and D be a database with distinct transactional data T_s . An inference in the form of $X \rightarrow Y$, where X, Y subset of I are sets of things called item sets, and $X \cap Y = \phi$. X . The rule states that X implies Y when X is antecedent and Y is consequent. Support(s) and trust are two fundamental basic metrics for association rules (c).

Support(s): An association rule was defined as the proportion of data in the system that contain $X \rightarrow Y$ compared to the total number of entries. During the scanning procedure, the number for each item is incremented by one every time the item is discovered in a different operation T in data D . It signifies that the support number does not account for the item's amount. For example, if a consumer purchases three pints of liquor, we only boost the beer's support count by one; in another terms, if a transaction involves an item, the item's support count is raised by one. The formula for calculating support(s) is as follows:

$$Support(X \cup Y) = \frac{Supportcountof X \cup Y}{TotalnumberoftransactioninD}$$

Confidence(c): An association rule was defined as the ratio of the number of accounts containing (X Y) to the total number of entries containing X, with the proportion over the confidence level generating an interesting connection rule X Y. Equation:

$$Confidence(X \cup Y) = \frac{Support(X \cup Y)}{Support(X)}$$

Confidence was an indicator of the magnitude of association rules; for example, if the confidence of the linear regression X Y is 79%, it indicates that 79 percent of the transactions containing X also contain Y. Users can also choose a minimum confidence level to ensure that the rules stated are interesting.

6.1.10. Distributed Association Ruling

DARM extracts rules from a variety of spatial datasets in a distributed context [5]. In contrast to a distributed system, a parallel data transmission does not have quick communication. As a result, distributed mining typically seeks to reduce transmission costs. To mine principles from dispersed datasets partitioned over three separate sites, researchers used high-speed DMA. FDM identifies the community support counts at each location and prunes all uncommon ones. Following the completion of home trimming, each site sends out signals to the other websites, requesting their contribution counts. It next determines whether or not large item sets are globally common, and builds candidate unit sets from such globally item sets.

6.1.11 GIS System Statistical Analysis

Geographic data now is employed in a variety of applications, including urban development management, transport improvement, communications and advertising, and so on. Normally, GDBMD collects and manages geographically useful information. Although several new techniques are developed that give operations and functionalities for geographical data analysis, they are inefficient for huge databases since GIS cannot find undiscovered knowledge. This type of information, which is the foundation of the KDD, must be developed using specialised approaches. Data mining is a method for extracting relevant information from large databases. The first purpose for retrieving information from the server is to make a forecast, and the next goal is to make a description. For data mining from a data, there are various mining methods available, such as ARM, grouping, and categorization; among these, the SARM idea is utilised in the geographical region, therefore the concept is spatial cluster analysis mining, in which data is extracted from geographic regions.

The association rule mining idea is used to estimate the common item set in a dispersed environment and determine the relations between various attributes by using the threshold level of trust and confidence. We separated the entire territory into three distinct regions, each with its own spatial data SDB1, SDB2,..SDBn and core values SK1, SK2, SKn, or Select N number of regions, each with its own data SDB1, SDB2,, SDBn. Each region determines their most common item sets as well as the worth of their support. The half support is found once each region is placed in a ring design. Now, area 1 sends its limited explanation (PS) value to region 2, and region 2 transmits its value to region 3, and so on until region n gives its value to region 1, at which point region n sends its value to region 1. Region 1 determines their real support by subtracting all of the Random Number values from the Partial Support values. Now, region 1 transmits the current support level to the entire dispersed environment's area.

6.2 GREEN COMPUTING'S ASCENSION IN THE MODERN COMPUTING ENVIRONMENT

Multiple utility-based apps can be conducted in today's computer environment, such as recovery and backup , which is critical in a cloud service where many systems perform their

tasks and duplicating structure is unnecessary. SaaS, on the other hand, is a cloud computing technique. There are instances when providing apps as a service stands to reason, if it's a payroll or CRM system. Frequently, the internal IT organisation lacks the competence needed to run a specific application, or the implementation is not important enough to support allocating limited IT capabilities to its management [9, 10]. There's no denying that cloud computing has security dangers, but as with anything else in life, the risks must be evaluated against the possible advantages.

Algorithm . Encryption Process**BEGIN**

Step 1: Take the Spatial Database

Step 2: Convert into the horizontally partitioned distributed database (N Number of datasets)

Step 3: Calculate the support count of each database.

Step 4: Calculate the support and confidence.

Step 5: Calculate partial support and partial confidence.

Partial Support (PS) = X. Support - DBMinimum Support

Partial Confidence (PC) = X. Confidence - DB x Minimum Confidence

Step 6: Add their own private key in all partial support and partial confidence.

Partial Support(PS) = X. support - DBminimum support + Key

Partial Confidence(PC) = X. Confidence - DBxMinimum Confidence+Key

Step 7: Divide the partial support and partial confidence into the three different values.

Step 8: Convert partial support, partial confidence and partial lift values into the ASCII value and compute the matrix Y.

Step 9: Take the transpose of the matrix (YT).

Step 10: Exchange YT into the Binary format.

Step 11: Let own key matrix X

Step 12: Exchange X into binary

Step 13: Execute Ex-or among X and Y.

Step 14: The matrix (Step 14) stored in associate memory.

Setp 15: The resultant matrix is sanded to the protocol initiator Server.

END

Algorithm . Decryption Process**BEGIN**

Step 1: Let encrypted matrix M

Step 2: Calculate transpose of M into MT

Step 3: Exchange MT into binary

Step 4: Let own key X (Matrix)

Step 5: Exchange X into binary

Step 6: Execute Ex-or among MT and X

Step 7: The result (Step 6) is converted to the ASCII code (Original Matrix).

Step 8: After receiving all the original values from the different databases, the protocol initiator takes the step for data analysis by calculating Global support and confidence.

Step 9: After that, the protocol initiator broadcasts the results to all the database server admin present in the distributed environments.

END

More tailored app services are likely to be the next huge concern in cloud computing. Many IT firms are unable to afford computer chip infrastructure. However, access to some quite compute-intensive analytic apps could be beneficial to the company. All of this does not imply that on-premise apps and architecture will be phased out. On a pragmatic level, there are just too many existent apps that cannot be redesigned to operate on a cloud platform in a cost-effective manner. On a tactical level, hundreds of apps are too critical to the business to function in the cloud. Lastly, a number of regulatory and legal obstacles may prevent cloud computing from becoming practical in some circumstances [6]. Cloud computing isn't a one-size-fits-all solution. We're gradually moving toward a mixed computing paradigm that combines the finest features of cloud services with on-premise apps that operate on internal IT networks with similar designs to public cloud services. And once that occurs, we'll be entering an unique age of IT flexibility, where IT organisations will be able to dynamically react to the business's quickly changing demands for the first time, rather than always attempting to get the industry to adapt to the way IT operates.

Cloud Computing Abuse and Malicious Use: Many cyber thieves are attracted to IaaS solutions because of the simplicity of registration and relative anonymity they provide. Botnets and their management & control centres have been reported to be hosted on IaaS platforms, as well as distribution for vulnerabilities, trojans, and other malware. In-the-cloud abilities can be abused in a variety of ways; potential future applications involve launching static attack points, CAPTCHA-solving farms, credential and key decryption, and more. To address this, IaaS providers should strengthen the weakest links: the signup process and client network traffic management.

Customers engage with cloud services through software endpoints or APIs, which must have exceptionally mutual communication, authentication, cryptography, and integrity verification mechanisms, particularly when third parties begin to build on them. A careful examination of the interfaces and quality execution of the security procedures are the keys to fixing those challenges.

Data Loss or Leakage: Data can be stolen or lost in a variety of ways, including removal without a restoration, loss of the encryption key, or unauthorised access. This is one of the most pressing worries for organisations, because they not only risk losing their status, but they are also legally compelled to protect it. To prevent such incidents, a variety of measures can be taken, ranging from the continuous use of encryption and high-quality disaster restoration to contractual requirements for backup and private disposal processes.

Hijacking an account or service allows an attacker to receive data, modify data, falsify payments, and reroute your customers to unauthorised websites. In today's world, all it takes is a legitimate phishing page or an excellent interpersonal engineering strategy to hand over the keys to your kingdom. This should be avoided via identity verification procedures, security regulations, and monitoring.

Uncertain Risk Profile: At all times, security must be at the top of the priority list. Updates to code, security procedures, security profiles, and intrusion attempts must all be kept in mind at all times.

6.3 GREEN COMPUTING

Green computing was becoming increasingly popular as energy costs rise and environmental issues grow. Design and hardware architectures have been studied for performance, dependability, reliability, and privacy in both computing and communication systems. However, there has been minimal effort on analysis depends on the amount of resource that the Processor will require. Because most communication systems must operate 24 hours a day, seven days a week (for example, most server farms and servers in a cloud computing platform), the power consumption of a system focusing on a particular system design is critical. For example, a system that consumes a lot of energy will always have a higher operating cost. High power consumption also means that more energy is generated, necessitating the use of more power to cool down. Global warming, which is generated by carbon releases, is today's largest environmental concern. The energy problem has given rise to the idea of green computing, which necessitates the redesign of algorithms and systems to be more energy efficient. Green IT is the study and practise of utilising information resources in a way that is effective, efficient, and cost-effective. Virtualization, electricity management, resource recovery, and telecommuting are some of the green IT initiatives. The core premise of cloud computing is that computation is distributed among a large number of servers rather than being done on a single machine or on a remote server. Edge computing, distributed systems, and parallel computing are all extensions of cloud computing. Its specialty is secure, rapid, and easy data management and network computing services based on the Internet. A huge number of virtualization currently squander a significant quantity of energy and emit a significant amount of carbon dioxide. As a result, major reductions in emissions and waste consumption are required. Both private and public clouds are considered in the analysis of energy consumption in cloud computing. Cloud computing with a green method can make processing power more energy-efficient [8]. Green computing was the science and practise of efficiently and successfully planning, producing, utilising, and discarding of computers, processors, and other subsystems like displays, printers, disk drives, and connectivity and wireless communications with low or no environmental impact. [9]. Green computing can be approached in a variety of ways, including:

1. Product durability

2. Effectiveness of algorithms

3. Allocation of resources

4. Virtualization

5. Power control

6.4 SUMMARY

Different research findings were discussed in this chapter that are highly valuable for real-world applications, such as KVM, Xen, and the advent of green computing in the cloud. Finally, this chapter focuses on a single case study that is particularly valuable for statistical analysis in remote settings. There are many methods for either relational or geographic datasets that have been proposed to replant the frequent item sets and classification methods; here, an algorithm has been proposed to find optimal spatial association rules, which is solely portrayed in GIS data models and geo-ontologies by one-to-one and one-to-many interactions. The purpose of this chapter was to describe an approach for improving spatial association rule mining. There are three basic steps in the suggested method. For starters, it simplified the GIS module's geographical information preprocessing chores. The next step is to remove all well-known GIS dependencies that compute the link between several properties. Finally, when there are more than two segments, a method was explored in this study to give the highest level of privacy, with each discovering a clustering algorithm between them with 0 percent security breaches.

REFERENCES

1. Moschakis IA, Karatza HD. Evaluation of gang scheduling performance and cost in a cloud computing system. *The journal of supercomputing*. 2012 Feb;59(2):975-92.
2. Dash M, Mahapatra A, Chakraborty NR. Cost effective selection of data center in cloud environment. *International Journal on Advanced Computer Theory and Engineering (IJACTE)*. 2013;2:2319-526.
3. Abirami SP, Ramanathan S. Linear scheduling strategy for resource allocation in cloud environment. *International Journal on Cloud Computing: Services and Architecture (IJCCSA)*. 2012 Feb;2(1):9-17.

4. Majumdar S. Resource management on cloud: handling uncertainties in parameters and policies. *CSI communications*. 2011 May;22:16-9.
5. Roy N, Dubey A, Gokhale A. Efficient autoscaling in the cloud using predictive models for workload forecasting. In 2011 IEEE 4th International Conference on Cloud Computing 2011 Jul 4 (pp. 500-507). IEEE.
6. Farooqi AM. Comparative Analysis of Green Cloud Computing. *International Journal of Advanced Research in Computer Science*. 2017 Mar 1;8(2).
7. Masoud RI, AlShamrani RS, AlGhamdi FS, AlRefai SA, Hemalatha M. Green Cloud Computing: A Review. *International Journal of Computer Applications*. 2017;167(9).
8. Piraghaj SF, Dastjerdi AV, Calheiros RN, Buyya R. ContainerCloudSim: An environment for modeling and simulation of containers in cloud data centers. *Software: Practice and Experience*. 2017 Apr;47(4):505-21.
9. Khosravi A, Nadjaran Toosi A, Buyya R. Online virtual machine migration for renewable energy usage maximization in geographically distributed cloud data centers. *Concurrency and Computation: Practice and Experience*. 2017 Sep 25;29(18):e4125.
10. Machen A, Wang S, Leung KK, Ko BJ, Salonidis T. Live service migration in mobile edge clouds. *IEEE Wireless Communications*. 2017 Aug 3;25(1):140-7.



Mrs. Srividhya Elangovan currently working as an Assistant Professor in the Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology. She received her B.Tech from Prince Shri Venkateshwara Padmavathy Engg College, master degree from Vinayaka Missions Research Foundation and submitted her PhD Thesis at Bharath Institute of Higher Education & Research, under the title “Diagnosis of Diabetes by Tongue Analysis using Image Processing”. She is having 10 years of teaching experience and 3 years of industry experience. Besides she is a life member in Indian Society for Technical Education and International Association of Engineers. Despite teaching the students, she has filed 3 patents, 11 Scopus publications in both international and national journals and also presented more than 20 papers in both national and international conferences. She is one of the reviewer in Journal of Pharmaceuticals Research International. She has written and published the book under the title Programming in Python in 2019.



Mrs. Jayanthi Sampath is an assistant professor of Computer Science and Engineering at Sathyabama Institute of Science and Technology. She holds a Master Degree in Software Engineering and prior to completing her PhD from Anna University. Her current research includes Bioinformatics integrated with Machine Learning. Her articles on analysis of gene microarray datasets have appeared in a number of journals and also indexed in web of science.



Ms. A. Sonya is currently working as Assistant Professor in Department of Information Technology at B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu. She had completed her M.E degree in Computer and Communication Engineering in 2012 and B.Tech degree in IT in 2010 from Anna University, Chennai. Currently, she is pursuing Ph.D in renowned University. She has 8 years of teaching experience. Her current research interests include Cloud computing, Data Structure, Cloud Forensics and Cyber Security. She has published more than 15 Journals and Conference papers. She has a published patents and book chapter. She has also received a certificate of grant innovation Australian Patent.



Dr. Nalini Subramanian currently working as an Associate Professor in Department of Computer Science and Engineering, Prathyusha Engineering College, Aranvoyal Kuppam, Thiruvallur. She has pursued her Doctorate (Ph. D.) degree in Department of Computer Science & Engineering at Sathyabama University, Chennai, India in 2020. She has received her M.E.(CSE) at Sathyabama University, Chennai, India in 2006. She has 15+ years of teaching experience. She has published more papers in International Journals and Conferences. She has published two patents in recent domains. Her area of research interest includes Cloud computing, Network security, Block chain, Machine Learning and Deep Learning.



Dr. S. Gokulakrishnan Completed his B.Tech (Information Technology) from Pallavan College of Engg , Kanchipuram, Tamil Nadu, India and M. Tech (Information Technology) from Sathyabama University, Chennai, in the year 2005 and 2011 respectively. He also completed his Ph.D (in the area of Cloud Computing) in Sathyabama University, Chennai. His areas of interest include Cloud Computing and Big data Analytics. He has got around 16 years of teaching experience in various Institutions. Currently working as a Assistant Professor in CSE Department at Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya , Enathur Kanchipuram.



**KALAIVANI
PUBLICATIONS**



9788195434305

Reach as

16, Weekly Market Road, Erode, Tamilnadu, India -638301

Phone No : 04256234667