

A Major Project Report On

**‘USING HIDDEN MARKOV MODEL TOWARDS SECURING  
THE CLOUD: DETECTION OF DDoS SILENT ATTACKS’**

Submitted in partial fulfillment of the requirements  
for the award of a degree of the

**MASTER OF TECHNOLOGY**

**IN**

**SOFTWARE ENGINEERING**

By

**Fungayi Donewell Mukoko**

(Roll No. 2k12/SWE/26)

Under the Guidance of

**Dr. Ruchika Malhotra**

Department of Software Engineering

Delhi Technological University, Delhi



**Department of Computer Engineering**

**Delhi Technological University, Delhi**

**2012-2014**

# Certificate

---

---



**Delhi Technological University**

(Govt. of National Capital Territory of Delhi)

Bawana Road, Delhi – 110042

Date: \_\_\_\_\_

This is to certify that the thesis entitled '**Using Hidden Markov Model towards securing the cloud: Detection of DDoS silent attacks**' done by **Fungayi Donewell Mukoko (2K12/SWE/26)**, for the partial fulfilment of the requirements for the award of the degree of Master of Technology in Software Engineering, is an authentic work carried out by him under my guidance. The matter embodied in this thesis has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

**Project Guide:**

**Dr. Ruchika Malhotra**

Assistant Professor, Department of Software Engineering

Delhi Technological University, Delhi 110042

## Acknowledgement

---

---

First and foremost, I would like to convey my sincere appreciation to my supervisor **Dr Ruchika Malhotra** for her patience, concern, invaluable guidance and also encouragement throughout the preparation of this project. Moreover I would like to dedicate my deepest thanks to my beloved family members for their moral support. Thanks to all the lecturers, friends, and fellow classmates for their cooperation and sharing. Last but not least, my thanks go to Nakul sir for his treasured assistance.

Fungayi Donewell Mukoko

2k12/SWE/26

M.Tech (Software Engineering)

4<sup>th</sup> Semester

# Table of Contents

---

---

<b>CHAPTER 1.....</b>	<b>1</b>
<b>Introduction .....</b>	<b>1</b>
1.1 Basics of the work .....	1
1.2 Motivation .....	2
1.3 Objectives and Goals.....	3
1.4 Organisation of Thesis .....	4
<b>CHAPTER 2.....</b>	<b>6</b>
<b>Literature Review .....</b>	<b>6</b>
2.1 Review Procedure.....	6
2.2 Summary of the Review Conducted .....	6
<b>CHAPTER 3.....</b>	<b>18</b>
<b>Research Background .....</b>	<b>18</b>
3.1 Cloud Computing Security Measures and Metrics .....	18
3.3 Empirical Data Collection.....	23
<b>CHAPTER 4.....</b>	<b>26</b>
<b>Research Methodology .....</b>	<b>26</b>
4.1 Methodology used.....	26
4.2 Explaining Cloud Computing.....	26
4.2.1 Participants .....	28
4.2.2 Cloud Services.....	29

4.2.3 Isolation Levels .....	30
4.3 The statistical model.....	31
4.3.1 Three elementary problems in tackling HMM .....	31
4.3.2 Remedies for these three basic problems .....	31
4.3.3 In-depth understanding of HMM.....	32
4.3.4 Markov Chains .....	33
4.3.5 Elements of HMM .....	34
4.4 Analysing Hidden Markov Models .....	36
4.4.1 Overall implementation demonstration of HMM in the cloud .....	38
4.4.2 Essentials of Hidden Markov Model .....	41
4.4.3 Properties of the Markov Process .....	41
4.4.4 Probabilistic Inference .....	41
4.4.5 Viterbi Algorithm .....	47
4.4.6 Forward $F(k,i)$ and Backward $B(k,i)$ Probability Calculation .....	49
4.4.7 HMM Training (Baum-Welch Algorithm) Counts .....	51
4.4.8 Interplay between Two Equations (Expectation Maximisation) .....	53
4.4.9 Computational Part .....	54
4.5 Description of the proposed framework.....	55
4.5.1 Resembling the attacker scenario .....	55

4.5.2 Design, Modelling and Experimentation of the DDoS Silent attack detection framework.....	56
4.6 Performance Evaluation Measures.....	59
<b>CHAPTER 5.....</b>	<b>61</b>
<b>Result Analysis.....</b>	<b>61</b>
5.1 The estimate of a statistical model according to a training set.....	61
5.2 New training data sampling, sequence probability and probable states path.....	63
5.3 Discussion of Results .....	64
<b>CHAPTER 6.....</b>	<b>72</b>
<b>Conclusion and Future Work.....</b>	<b>72</b>
6.1 Summary of the thesis .....	72
6.2 Discussion of Results .....	74
6.3 Application of the Work.....	74
6.4 Future Work .....	75
<b>References.....</b>	<b>76</b>

## List of Figures

---

---

Figure 1: Goal-Question-Metric (GQM) framework .....	23
Figure 2: NIST visual model of cloud computing definition .....	27
Figure 3: NIST cloud service model scope and control .....	28
Figure 4: The Markov Model's State Diagram .....	33
Figure 5: An unfolded HMM in time as a lattice (or trellis) .....	35
Figure 6: Implementation demonstration diagrammatic representation.....	38
Figure 7: The combined compacted probabilities (Joint event/ Joint probabilities) representation.....	39
Figure 8: Viterbi Algorithm for detection problem (first two symbols).....	40
Figure 9: Probabilistic Finite State Machine (FSM) implementation (Viterbi Algorithm Illustration) .....	44
Figure 10: Developing the tree of the FSM.....	45
Figure 11: Developing the tree continued.....	46
Figure 12: Key Intuition .....	51
Figure 13: Baum-Welch Algorithm: Counts .....	52
Figure 14: Resembling the attacker scenario.....	56
Figure 15: Proposed Framework .....	57
Figure 16: A graphical representation of the selected data obtained from the selected simulation/experimentation number .....	66
Figure 17: Initialisation of the prior, transition and observation matrix as given from the selected simulation number .....	67
Figure 18: Iterations with their corresponding estimated log like-hood .....	67
Figure 19: Obtained results after Iterations .....	68

Figure 20: The obtained “Expectation Maximisation” training graph, drawn from the first simulation/ experimentation number .....68

Figure 21: The obtained best parameter view from the simulation.....69

Figure 22: Sequence 1 along with its hidden states, given a hackers trend used .....69

Figure 23: Sequence 3 along with its hidden states, given a hackers trend used .....70

Figure 24: Sequence 755 along with its hidden states, given a hackers trend used .....71



## List of Tables

---

---

Table 1: Literature Review .....	7
Table 2: Metrics definition .....	20
Table 3: Cloud Security Framework Metrics Definition.....	22
Table 4: Scenario explanatory Cloud Tracing Tabulated form .....	24
Table 5: Probabilistic checked with Logical Inference .....	42
Table 6: Tabular representation of the tree.....	46
Table 7: Table of counts .....	52
There are two main types of learning, as given in table 8: .....	61
Table 8: Learning comparatives .....	62
Table 9: Simulation Possibilities defined by Silent attack number along with Observables .....	65
Table 10: Available simulation/experimentation number against the total number of sequences .....	65

## Abstract

---

---

Cloud Computing has presented itself as a promising solution to new entrepreneurs as well as existing organizations for management of their IT needs at various levels. Many cloud service providers have exposed cloud services at cheap prices, which allow users at all levels of society to materialize their ideas and make them available across the globe. While the response has been overwhelming, the application areas where security of data is of utmost importance have not shown much interest. Hence incorporating dependable security measures in the cloud computing technology would be a good move since the aspect of security has turn out to be one of the main things to consider.

In this thesis work we took an initiative as we adopted and/or interpolating Hidden Markov Model into the circles of Cloud Computing, as we exploited it's capabilities in detecting the silent attack traces that dodge and/or bypass a set of methodical mechanisms intended to sense and prevent them into the cloud system. We modelled our hackers' attack scenarios where we defined some states and observations. The hackers' clusters have been taken to be the states, along with the observations which have been taken to a series of Virtual Machines on to which the attacker trades upon from the silent invasion through to the vulnerable and finally the target Virtual Machine. The modelling is given in such a way that, if we have a generated sequence of observed attacks, we must be able to find out the roots of done attackers from the grouped hackers.

# CHAPTER 1

## Introduction

### 1.1 Basics of the work

Cloud has form of equivalent and/or analogous as well as dispersed structure that consist a pool that has virtualised and inter-allied computers; dynamic in nature, and can be provisioned and presented as a single service level agreement put into place through negotiations between two parties which are the service provider and clients. Cloud computing is described as such technology that provides handy, on-demand access to pooled computing resources which can be configured, for instance servers, networks, storage facilities, along with applications rapidly provisioned then released with minimum administration effort [1]. Cloud Service Providers (CSPs) offer their services in numerous elementary models, which are infrastructure as a service, platform as a service, as well as software as a service.

Technically, cloud computing technology is supported by virtualisation technologies as well as concepts that necessitate pool of resources to be accessed in a shared manner. Physical IT resources are divided into logical units that are made to be accessible and available to various customers and allow simultaneous utilisation of resources through the use of these virtualisation technologies. This means that a number of logical clients can be served by means of a shared infrastructure, so that optimised use of cloud service vendor's infrastructure exists. Cloud computing is a suitable technology to meet the demands of the people. It entails sharing of various computing resources so as to handle applications, thus this technology migrate application software along with databases to huge data centres, and in this instance there is remotely sharing of computing resources. However this poses many security challenges which are have been tried to be counter-attacked by a number of

security approaches and techniques (in their variety) as given as part of this thesis work, from where we have derived the basis on to which we grounded our work, as a way of improving cloud computing environment's security aspects, factually it's because cloud security mandatorily needs consideration. In our thesis work we take the scenario of the attacker invading the cloud system by persistent silently attacking the cloud system through silent illegal Port scanning, thereby getting accessing into the system environment. The hacking program/hacker then attacks the vulnerable VM, and it continuously passes on to the next potential victim and it reaches its target VM. In this process there will be DDoS as the hacker floods persistent and multiple requests from the vulnerable till the target VM. So the hacker leaves a sequence an attack trace (which we intend to detect) as it performs DDoS attacks after a silent invasion.

## **1.2 Motivation**

Cloud Computing has presented itself as a promising solution to new entrepreneurs as well as existing organisations for management of their IT needs at various levels. Many cloud service providers have exposed cloud services at cheap prices, which allow users at all levels of society to materialize their ideas and make them available across the globe. Hence seeing the benefits that are being outstretched by cloud computing, and knowing that security is the key note which is a considered factor that has to be taken into cognisance by any organisation that intend to use the cloud for its beneficial use, we have been motivated to take a look into cloud security at a closer range. Furthermore, it has been seen that some tactical hackers are using silent attacks to invade into systems, where a silent (or stealth) attack refers to such attacks (be it an event, object, or file) that dodges and/or bypass a set of methodical mechanisms that are put in place to sense and prevent it,

without raising alert triggers. These silent attacks are characterised by low observability along with bad traceability. Illegal port scanning followed by instant invasion is an instance of this. [23] [54]. Distributed Denial-of-service (DDoS) attack accolades numerous compromised systems (multiple compromised VMs) attacking the particular target (target VMs), in so doing causing a detrimental denial-of-service to authorised clients/consumers of a Cloud Service Provider (CSP) of the targeted system Cloud Service Provider (CSP). The flood of incoming messages (silent persistent attacks) forces the VM to close down. So to give out a remedy to this, we tend to employ Hidden Markov Model (HMM), which has the capacities of modelling sequence of attacks left by the attacker or hacker (unseen events), hence we dwelt on this aspect of “using Hidden Markov Model towards securing the cloud: detection of DDoS silent attacks”. Hidden Markov Model in particular, is composed by a family of algorithms, which are Viterbi and Baum-Welch (forward-backward) algorithms and has its powerful properties which we need to tape into the cloud environment

### **1.3 Objectives and Goals**

The aim of this thesis is to achieve the following goals:

- To establish the impact of Machine Learning algorithms, Hidden Markov Model in particular, harnessing its characteristics in offering security features in the cloud computing environment.

The powerful nature of Hidden Markov Model, will be shown as we derive the appropriate states (blindly unseen events) from the given observation sequences (provided limited knowledge of the problem), in this way it gives the actual

pictorial view of the unseen events that happens in the cloud computing environment where no client has the full control of the used cloud resources.

- To simulate/experiment on the effects of Hidden Markov Model in detecting the DDoS attack trace by means of giving a series flooded Virtual Machines (VMs) by the hackers/attackers so as to guarantee the usefulness of our intentions.
- The main goal is to monitor VM cloud requests activities, determining their source IP addresses (point of entry, accessed after illegal port scanning) and the activities that they would have performed in the cloud system (flooding of persistent multiple requests). After designing the model, with the help of testing data, we evaluate the performance of designed model.

#### **1.4 Organisation of Thesis**

The remainder part of the thesis is ordered in the following chapters:

##### Chapter 2: Related Work

The section shows the summary of the work done by different researchers along the lines of securing the cloud. The final conclusions made by studying various papers from the year 2006 to 2013 are as well given in this chapter.

##### Chapter 3: Research Background

This section describes the background of this research work in detail. Dependent and independent variables along with the dataset used are given this chapter, strictly aligned to the security aspects of cloud computing.

##### Chapter 4: Research Methodology

This chapter provides the research methodology, covering all aspects of the method, measures of performance and giving the in-depth details of framework of the developed tool, as in according to the security paradigms in cloud computing.

#### Chapter 5: Results Analysis

We evaluate and judge the performance of our results in this chapter. We explain and make an evaluation of the resulting obtained results. Checking on how well the protection is guaranteed in the cloud computing environment.

#### Chapter 6: Conclusion and Future Work

Conclusions are drawn here. This section also incorporates the scope of future integration.

#### References

This section gives the reference details used in this study. Included in this material are the research papers along with books and URLs used and studied.

## **CHAPTER 2**

### **Literature Review**

There is always a petition to produce superb security techniques to secure the cloud, and as it is given in this chapter the flow of the efforts that have been done can be tracked. The section shows the summary of the work done by different researchers along the lines of securing the cloud. Diversity in angle and/or level in a way to secure the cloud is shown in these cloud security techniques, when they vary in their securing methods, in methods deployed by innumerable authors.

#### **2.1 Review Procedure**

In our review, we have considered only those papers where security techniques to secure the cloud are used, procedurally: We thoroughly searched from reputed sources, collected and studied sound papers around security paradigms in cloud computing outputting significant review as summarized and indexed in Table 2.1.

#### **2.2 Summary of the Review Conducted**

We have undertaken a specific systematic review (SLR) [50] to discover published papers related to the present tense security techniques as we made a walk through on the security paradigms in cloud computing, in the selected the papers published between 2006 and 2013. This gives the true reflection of the security techniques used in the current world in Cloud Computing. We followed different steps in Systematic Literature Review. The review process phases are illustrated in table 1 as follows:



**Table 1: Literature Review**

<b>Indexing</b>	<b>Ref. No:</b>	<b>Year</b>	<b>Security Techniques</b>	<b>Description</b>	<b>Impact</b>
1	[2]	2009	homomorphic token along using distributed verification protocols for erasure-coded data	This scheme attains combinations of storage correctness assurance together with simultaneous data error localization, that is, the detection/identification of the server(s) that are misbehaving. Furthermore it supports secure as well as competent dynamic operation on data blocks.	Privacy, Security
2	[3]	2009	Database Service Provider (DSP) re-encryption technique	This approach implements the data's access control that is encrypted by the DSP in a selective manner; moreover it can relieve users from the key derivation procedure that is	Access control enforcement management

				complex.	
3	[4]	2010	RSA algorithm	It works as a mechanism to warranty cloud data security and encrypting the data whilst it is being transferred on the network. RSA is specifically used for public/private key generation as well as encryption	The technique resolves security and authentication problems thus it provides privacy
4	[5]	2010	Trusted Platform Module (TPM)	The method builds trusted computing environment tailored to a cloud system through incorporation of a trusted computing platform within cloud computing system.	It takes care of authentication, confidentiality and integrity
5	[6]	2010	Role Based Access Control Model (RBAC)	A two phase access control policy is employed at the API level. Any client has to be authenticated before accessing any resource, along with his/her	authentication

				credentials and attributes	
6	[7]	2010	two-way handshake scheme	It is based on token management that makes use of homomorphic token along with distributed verification for erasure-coded data	Storage correctness insurance plus data error localisation
7	[8]	2010	Single Information Owner (INO) as well as the (CSP) Cloud Service Provider secure communication channel	It is a secure communication channel where session keys and tickets are used.	trust and confidentiality
8	[9]	2010	homomorphic cryptography (Paillier scheme) together with Zero-Knowledge Proof	It is a privacy-preserving method for Cloud publish/subscribe facility	authentication, data integrity and confidentiality

			(ZKPK)		
9	[10]	2010	Cloud-based Service Security Lab	It is a virtualised testing environment specifically for security concepts and components that are related to a particular service. Moreover it necessitates monitoring plus analysis of various security configurations, concepts along with infrastructure components.	Security
10	[11]	2010	Mutual Protection for Cloud Computing (MPCC) functions	It is a mutual protection architecture. . Both the cloud provider and client security's profiles are matched so that a mutual acceptance can be attained on the whole security environment	Flexible authentication, authorization and control are the core aims
11	[12]	2010	Five security-related featured service deployment	Models to take care of the security concerns, addressing different security requirements and	Security

			models: Separation, Availability, Cloud Data Migration Service, Tunnel and Cryptography	scenarios	
12	[13]	2010	Hybrid tree model	It establishes an up to date security policy along with attributes and/or negotiation context facts	authentication
13	[14]	2011	hypervisor-based virtualization with supplementary security tools	It is a data security based on virtualization mechanism	Security
14	[15]	2011	Kerberos protocol	It exploits some collaborative trust model functionalities	Authentication and Authorization

15	[16]	2011	Attribute-based encryptions along with proxy re-encryption	This construction uses attribute-based encryption and/or some alternative with capacity of implementing spectacular access control like predicate encryption plus proxy re-encryption	Security
16	[17]	2011	Security Cloud architecture	It continuously actively monitor any policy violations by a Cloud Service Provider (CSP), and if there are any, a report is given, and then informed security decisions are taken	Security transparency
17	[18]	2011	3 Dimensional Security framework	It is a two staged framework consisting of data classification followed by priority rating	Data protection
18	[19]	2012	HyperSafe on Virtual Machine Monitor (VMM)	It provides high security levels by strengthening the VMM,	Integrity
19	[20]	2012	Cloud Networking Security	Permits users (clients) to define their security requirements, and they can latter on enforce	Security

			Architecture	them	
20	[21]	2012	Privacy Control Policy Enforcement	It is based on multi-level privacy policies incorporated to the data of the user and then enforced in the cloud on different levels (infrastructure and application)	Privacy
21	[22]	2012	Cloud Bursting Brokerage and Aggregation (CBBA) Algorithm	This algorithm is in four parts, where part one is based on C++ files, part two on Java files and part three on C# files. For each of these three parts, the algorithm unlocks the environment to perform aggregation plus bursting on the files for cloud1, cloud2 and cloud3 respectively	Secure sharing
22	[23]	2012	Hidden Markov Model	It analyses a number of logs (records) from the cloud system to mine the motives as per activities traced.	Attack trace

23	[24]	2012	Homomorphic encryption principles	It uses simple yet an effectual partial homomorphic encryption scheme to secure the Matrix-Vector product (MatVec) outsourcing	Security
24	[25]	2012	Hadamard matrix	Used for encryption plus decryption algorithms, used for the purpose of encrypting sensitive data	Security
25	[26]	2012	Homomorphic public key encryption design	This scheme allows the processing of data whilst it is encrypted. The security of this scheme is rooted on the difficult problem of resolving the two-element (PAGCD) Partial Approximate Greatest Common Divisor	Security
26	[27]	2012	robust and searchable encryption approach	This scheme also provides some fault-tolerant availability intended for cloud computing	Fault-tolerant availability
27	[28]	2012	Kerberos ticket-based	It is capable of removing unwarranted trust	Privacy



			scheme	around the circles of the Cloud Service Provider (CSP) plus provisioning accessibility for collaborators	
28	[29]	2012	Key Distribution Scheme	A protected outsourcing computation EXP ( $\alpha$ , u) is used as a central tool to warranty privacy inputs as well as outputs of the clients	Privacy, soundness and completeness.
29	[30]	2012	Obfuscation	Code Obfuscation is the custom of hiding the meaning, purpose and functions of the code away from attackers. It is used to mask the code's operations when executing in uncontrolled environments	Privacy
30	[31]	2012	Role Based Access Control (RBAC)	It is an access control mechanism protection mechanism	Access control

31	[32]	2012	Profiling-as-a-service architecture	The architecture gives sound security to the cloud by collectively detecting followed by filtering any undesirable traffic over and about cloud instances	Security
32	[33]	2012	Byzantine fault-tolerant cloud infrastructure	It guarantees the safety of the system and attains liveliness by covering and ultimately eliminating Byzantine defective nodes	Security
33	[34]	2012	Attribute-based access control	Given is a competent an encryption mechanism that has temporal access control for the cloud services through the aid of cryptographic integer comparisons as well as proxy-based re-encryption scheme in relation to present recent time	Access control
34	[35]	2013	Hidden Markov model	It takes care of intrusion detection for both	Attack trace

				inbound and outbound traffic	
35	[36]	2013	Private Circular Query Protocol (PCQP)	It simultaneously achieve location-based K-NN command plus to take care of privacy as well as accuracy concerns of privacy-preserving Location-Based Service (LBS).	Privacy
36	[37]	2013	HPISecure software prototype	It permits users (clients) to store up their data on the cloud in its encrypted version without violation of the application's functionalities	Data Confidentiality
37	[38]	2013	Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm	It guarantees the entire three-protection-method of verification, authentication as well as data security, just at the same point in time	Verification, Data Security, Authentication

## **CHAPTER 3**

### **Research Background**

Our study aims at incorporating security in the cloud computing environment. The foremost task is to select the cloud security metrics which are to be considered. In this regard, we present independent and dependent variables used in this study, dataset and data analysis techniques.

#### **3.1 Cloud Computing Security Measures and Metrics**

Measurement is undoubtedly a very important aspect of sound Software Engineering practices and in this regard, the “security paradigms in cloud computing” is not an exception. It follows from the notion that you cannot control what you cannot measure. Quite a number of researchers have proposed metrics for cloud security as a way to address the security concerns in cloud computing environment. Cloud computing has become a burning area of study in recent years, and this as well triggered cloud security concerns which in brought a consideration on the cloud security metrics which we will consider in this thesis work, as several researchers have given it in this domain. We further analysed the different perspectives of these cloud security studies.

Without taking consistent and objective measurements it is a challenge to demonstrate the performance of a Cloud Service Provider (CSP) and/or organization in matters of security [39]. Security metrics dashes quantitative measurement giving out the degree of trustworthiness in a Cloud Service Provider (CSP) [40]. Metrics can also be categorized as primitive or computed. Primitive metrics are those that can be directly observed whilst

computed metrics are those that are derived from computations coupled from certain metrics, of which cloud computing security metrics tend to fall in this category, because of the nature of the cloud [41], as follows table 2 gives some defined metrics and table 3 contains some cloud security framework defined metrics.

**Table 2: Metrics definition**

S.No.	Metric	Definition
1.	System's attack surface [42], [43]	Set of ways through which attackers penetrate systems with the potential of causing damage. It the subset of the resources of a system, encompassing data, channels, plus methods possibly used in system's attacks. Conclusively, the "smaller" an attack surface, the higher its system security.
2.	Average Active Vulnerabilities (AAV) coupled with Vulnerability Free Days VFD [44]	Available given two metrics report an outcome of the number as well as the frequency for new vulnerabilities being identified together with their lifespan.
3.	Common Vulnerability Scoring System (CVSS) plus Common Vulnerabilities as well as Exposures (CVE) [40], [45], [46]	CVSS encompasses Base, Temporal and Environmental metrics; delivers a tool that quantifies the severity coupled with the risk of vulnerability. CVE are identifiers, specifically, dictionary of common names aimed at holding information security vulnerabilities that is known publicly. They lessen the burden share data across dispersed network security tools as well as databases.
4.	Security Metrics Objective Segments (SMOS) [47], [48]	It is a model that systematizes and organizes security metrics development, and can be integrated with risk-driven security metrics development activities. The security measurement target in this study is a technical system.

5.	Survivability, Privacy, Confidentiality, Integrity, Availability, Accountability, Reliability, together with Non-repudiation [49]	A security metrics' set made to quantify several aspects of security for an organization
----	---	--

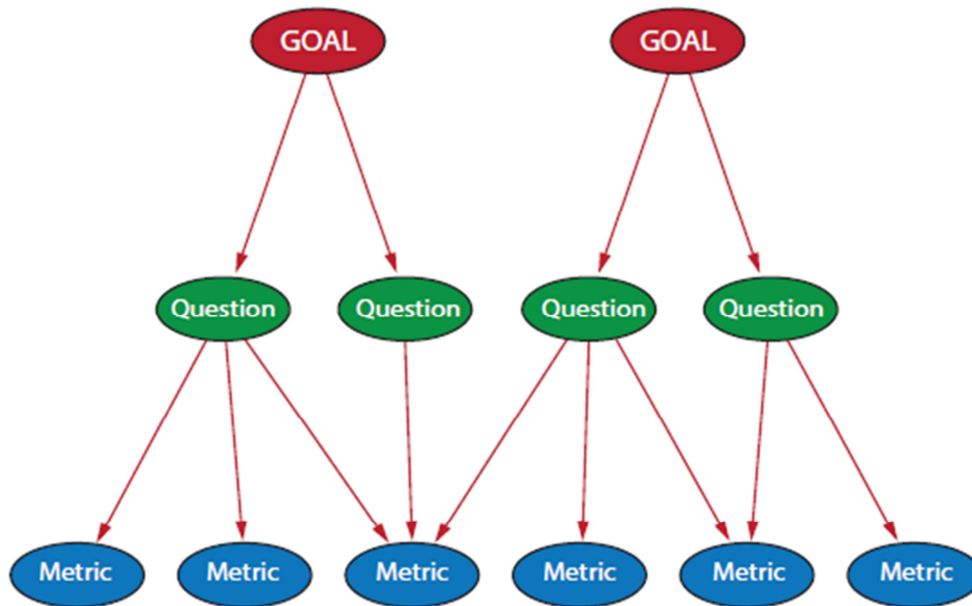
A proper as well as standard-based method for addressing security metrics for the Cloud still is a much tougher plus it's greatly an open issue [51]. For the purpose of validating (certifying) a security design, model-based approaches have been adopted by engineers [52]. Security measurement is yet still a fresh initiative; again some narrow and/or conclusive proclaims regarding such metrics would be, is debatable [53]. There is unavailability of universally plus well recognized metrics pertaining cloud security, so different cloud measurements that are appropriate to certain unique strategies and goals have been and can always be developed [53].

**Table 3: Cloud Security Framework Metrics Definition**

S.No.	Security Framework	Definition
1.	SPI security metrics framework for the Cloud[51]	It makes the assessment of Cloud Service Provider's security by a consideration of diverse service along with deployment models contained in the Cloud, and it then objectively and quantitatively measure the security of a Cloud service provider. (SPI stands for SaaS $\longleftrightarrow$ PaaS $\longleftrightarrow$ IaaS —security level of an authentication mechanism used via the IaaS' management interface)
2.	The Goal-Question-Metric (GQM) framework, as shown in figure 1	It is useful in describing strategy plus metrics for the complete cloud initiative, where need be. Triggered questions from the naturally set goals must without fail be answered as a way of determining the successfulness of the goal being met or not.



**Figure 1: Goal-Question-Metric (GQM) framework [53]**



### **3.2 Independent and Dependent Variables**

In this research, we have used seen that with cloud computing, there are a bit of some difficulties in clearly defining weather; variables are strictly independent or dependent variables.

### **3.3 Empirical Data Collection**

The user generated dataset with an indication of a good and real representation of what happens in the cloud computing environment has been used in this study. In out thesis work, the data is arrayed in a matrix form, for simulation execution purposes to show what takes place in the cloud environment. More so, it is iterated for different number of experimental cases.

**Table 4: Scenario explanatory Cloud Tracing Tabulated form**

<b>Indexing</b>	<b>Different Set of Hackers: (Attack Based on sources)</b>	<b>Sample size N (of Attacks Recorded: Randomized) in Cloud Environment</b>	<b>Vulnerable Silent Attack invasion (Easy target Virtual Machine)</b>	<b>Flooded Virtual Machines</b>
1	Set A	7324	VM <sub>i1</sub>	VM <sub>T1</sub>
2	Set B	8023	VM <sub>i2</sub>	VM <sub>T2</sub>
3	Set C	929	VM <sub>i3</sub>	VM <sub>T3</sub>
4	Set D	1181	VM <sub>i4</sub>	VM <sub>T4</sub>
5	Set E	580	VM <sub>i5</sub>	VM <sub>T5</sub>
6	Set F	858	VM <sub>i6</sub>	VM <sub>T6</sub>
7	Set G	1064	VM <sub>i7</sub>	VM <sub>T7</sub>

A trait (attack trace/ DDoS attack trace in virtual machine) is one which happens either in one variation or another, with no in-between. These invasion take place silently, letter appearing in the cloud system and shown silent attacks take place in very different forms (i.e. at different targeted virtual machines, flooding virtual machines of their interests at a particular point in time):

In our thesis work we studied in more detail the Cloud Virtual Machine DDoS with two and three silent attacks simultaneously. When we studied two silent attacks (as a way to give a pictorial view) we used 1 and 2, and three traits 1, 2 and 3 in the table 4 above.

## **CHAPTER 4**

### **Research Methodology**

In this section we present the methodology used. Performance evaluation measures are also presented. At the end, various validation techniques are explained.

#### **4.1 Methodology used**

In this study, we have first given a brief introduction about and gave a briefing on the standard definition of Cloud Computing and explain on its structure, to give an understanding on how we can incorporate our proposed model into the cloud architecture. Thereafter, we then dwelt on the model used, and we explained it in detail. The machine learning, statistical model contains a number of algorithms, which we each dismantled for necessary description.

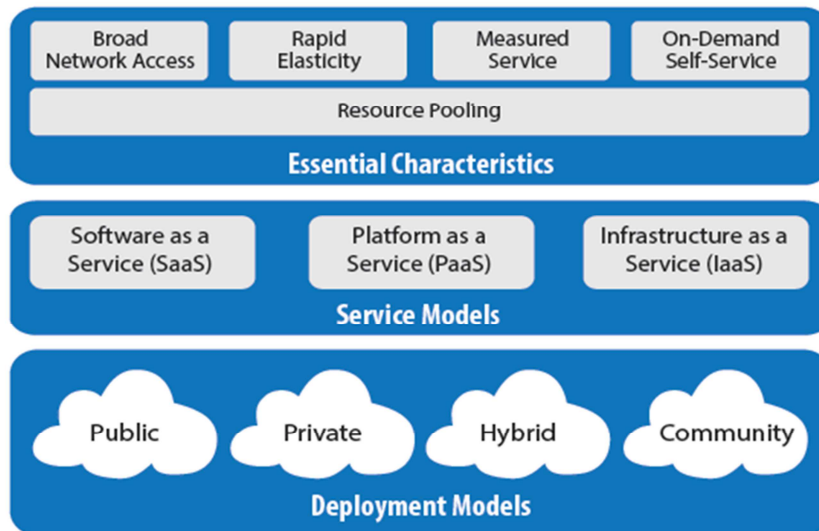
#### **4.2 Explaining Cloud Computing**

The taxonomy of clouds help us to better understand the scope of cloud computing and its related concepts and technologies and this will in turn give us a clue on how to incorporate our security techniques.

Many different views and definitions of cloud computing probed up during the course of the years, however the most dependable definition that we rely on is that given by the NIST, given like: “Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released

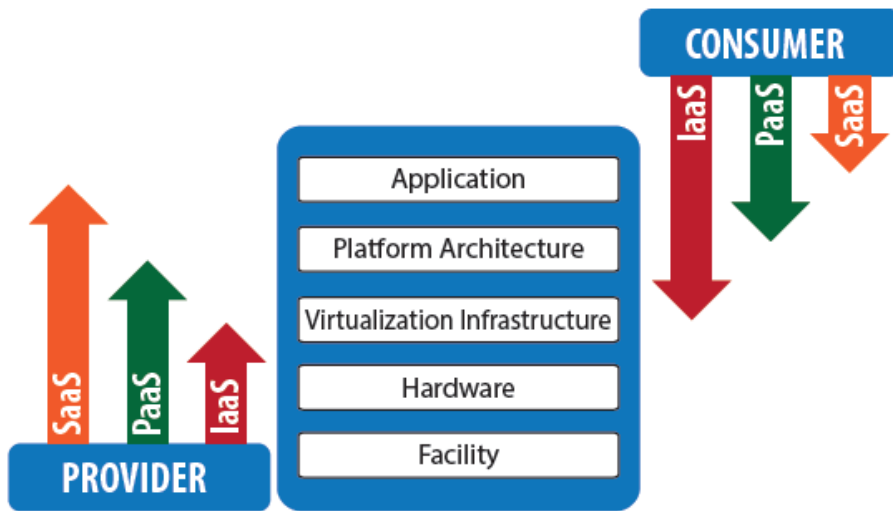
with minimal management effort or service provider interaction” [27]. A visual view is shown in the upcoming diagram, which is figure 2.

**Figure 2: NIST visual model of cloud computing definition [53]**



With the security analytical way of seeing, private cloud has almost the same weighting with traditional IT environments, holding the most control. In the service model, a cloud service provider holds control of the facility layer (specifically data center), along with necessary utilized physical hardware. Cloud consumers’ control increases with a certain margin in cases where service models move up the hierarchical stack from IaaS up to SaaS.

**Figure 3: NIST cloud service model scope and control [53]**



#### 4.2.1 Participants

In a cloud-model there are four main participants:

- i) Cloud Provider: A cloud provider (service provider) is an entity that is responsible for everything required for making a cloud service available.
- ii) Cloud Consumer: A cloud consumer is either a cloud service owner or a cloud service consumer. Cloud service owner is the individual or organization who subscribes for a cloud service. If there is any charge associated with the service, the cloud service owner will be responsible for the bills. Cloud service consumer is an individual or application who accesses a cloud service.
- iii) Cloud Broker: It is such entity that mediates between cloud providers and cloud consumers. The goal of a service broker is to provide the cloud consumer a service that is more suitable for its needs. This can be done by simplifying and improving the service and contract, aggregating multiple cloud services or providing value-added services. One can consider cloud brokers as a special cloud provider.

iv) Cloud Auditor: A cloud auditor is an independent party who examines a cloud service stack to provide an assessment on security, privacy and availability level of the corresponding cloud services and ensures that the corresponding SLAs (Service Level Agreement) are fulfilled. The details and scope of auditing process is normally specified in the service contract.

#### *4.2.2 Cloud Services*

The services provided by cloud providers can be divided into following three main layered categories, as given in figure 3. Each layer consumes services provided by the layer below it.

- 1) Software as a Service (SaaS): All types of softwares including financial, CRM, HR, Sales, and office assistance can be delivered as a service. Salesforce.com, Google Docs, and Zoho Docs are some examples of SaaS services. Consumers of SaaS services, who are usually end users of the application or software administrators, access these types of softwares through web browsers or mobile apps.
- 2) Platform as a Service (PaaS): Database, middleware, also integration bus are examples of platform resources which are delivered by PaaS providers in per service form. PaaS services are normally consumed by developers, testers, deployers, middleware/integration engineers and application administrators. As an example, Google App Engine is a popular Platform as a Service (PaaS).
- 3) Infrastructure as a Service (IaaS): IaaS clouds deliver their consumers with low level infrastructure resources, such as storage, Content Delivery Network (CDN), computational power, networks, backup and recovery, as a service. Typical IaaS

consumers consist of system developers, network engineers, system administrators, monitoring engineers and IT managers.

#### *4.2.3 Isolation Levels*

With respect to deployment model and isolation levels, clouds can be categorized into the following five categories:

- **Public Cloud:** A public cloud is a cloud that its infrastructure is shared by many mutually untrusted cloud consumers.
- **Private Cloud:** If the infrastructure of a cloud is dedicated to a specific organization, we refer to that cloud as a private cloud. A private cloud can be on or of premise.
- **Virtual Private Clouds:** Of-premise clouds that are isolated from untrusted organization only through virtual network isolation (not physical network isolation) are called virtual private cloud.
- **Community Clouds:** Community clouds are clouds that their services are accessible to a particular set of organizations which form a community. Community clouds can all be on or off premises.
- **Hybrid Clouds:** A cloud that is a composition of two or more types of clouds is called hybrid cloud. These types of clouds are becoming increasingly more popular. Integration of these clouds poses some security.



### 4.3 The statistical model

The statistical model under discussion is the Hidden Markov model (HMM) that we paid attention to, and checking how this can be well incorporated in the cloud computing environment to render the most needed security that has been addressed as a cause for concern. HMM is  $\lambda = (A, B, \Pi)$ , where A, B and  $\Pi$  are state transition and observation probability matrices, and starting state probabilities respectively.

#### 4.3.1 Three elementary problems in tackling HMM

- (1) With a pool of observations in addition to HMM, work out the observation sequence's probability.
- (2) With a pool of observations in addition to HMM, work out an best (hidden) state sequence.
- (3) With a pool of observations, search for peak state transition probabilities as well as observation probabilities.

#### 4.3.2 Remedies for these three basic problems

- i) Problem (1) is tackled by forward algorithm;
- ii) Problem (2) is tackled by Viterbi algorithm;
- iii) Problem (3) is tackled by the Baum-Welch algorithm; it as well achieves model training and parameter estimation.

### *4.3.3 In-depth understanding of HMM*

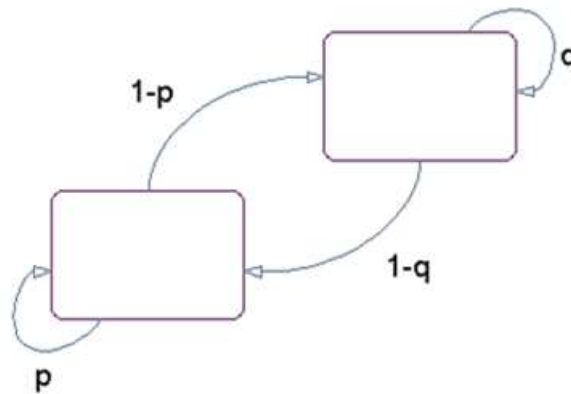
The power of HMM to be able to model input sequences as they are generated by a parametric random process triggered its use in this proposed model that have been implemented. So its goal is to characterize/ describe a sequence of events generated by a parametric random process i.e. a random process with certain parameters.

Hidden Markov Model is derived from the basic concepts of Markov Model in general, hence for a pictorial view of Hidden Markov Models, there is need for a brief walk through on Markov Models. In this regard, Markov models are used model input sequences that are generated by a parametric random process with the intention or goal to characterise these sequence of events, hence this feature is extendable to HMM.

#### 4.3.4 Markov Chains

The Markov model is set in a visual depiction in a way of such state diagram, in figure 4 as follows:

**Figure 4: The Markov Model's State Diagram**



The process that is being modelled is represented by rectangles which give possible states, as transitions between states, are given by arrows. The transition probability is tagged on each arrow. Stepwise in the process, there is possibility that the output or emission is generated by model, conditioned by the state in which it's in, and then there can be a transition of states. Markov models note-worthy characteristics: The next state independent of all other previous states, save for the current state, even those that resulted in that current state. Markov chains are considered as mathematical portrayals of Markov models that have set of states which are discrete, and are characterized through:

- 1) Set of states as  $\{1, 2, \dots, M\}$
- 2) Transition matrix  $T$  of dimensions  $M$ -by- $M$  with  $i, j$  entries which are a probability of state transitions from  $i$  to  $j$ . Row entries of  $T$  should mandatorily add up to 1, and this will in turn fulfil the law probabilities, which says probabilities must sum up to 1 as we dealing with transition probabilities.

- 3) A set of potential outputs, or emissions, i.e.  $\{S_1, S_2, \dots, S_N\}$
- 4) An emission matrix  $E$ ,  $M$ -by- $N$ , whose entries  $e_{i,k}$  offers the probability that symbol  $s_k$  can be emitted assumed that the model's current state is  $i$ .

Now delving on hidden on Markov Model, we can specifically define it as special type of Markov Model.

In Hidden Markov Model (HMM), there are two sources of randomness.

- i) Randomly moving from one state to another,
- ii) The observation in a state is also random

#### 4.3.5 Elements of HMM

In a nutshell HMM elements are:

1.  $N$ : A number of a model's states

$$S = \{S_1, S_2, \dots, S_N\}$$

2.  $M$ : Totality of ordered number of standalone observation symbols.

$$V = \{v_1, v_2, \dots, v_M\}$$

3. Transition probabilities for states:

$$\mathbf{A} = [a_{ij}] \text{ such that } a_{ij} \equiv P(q_{t+1} = S_j | q_t = S_i)$$

4. Observation probabilities:

$$\mathbf{B} = [b_j(m)] \text{ where } b_j(m) \equiv P(O_t = v_m | q_t = S_j)$$

5. Opening state probabilities:

$$\mathbf{\Pi} = [\pi_i] \text{ where } \pi_i \equiv P(q_1 = S_i)$$

N as well as M both are contained in the other strictures consequently

$\lambda = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$  a group of parameters for HMM.

The main goal is to determine a model's parameters, having granted training set containing sequences. Naturally numerous diverse state sequences Q can produce similar observation sequence O, but having different probabilities but we are interested in the one having the highest likelihood in producing the sample.

**Figure 5: An unfolded HMM in time as a lattice (or trellis)**

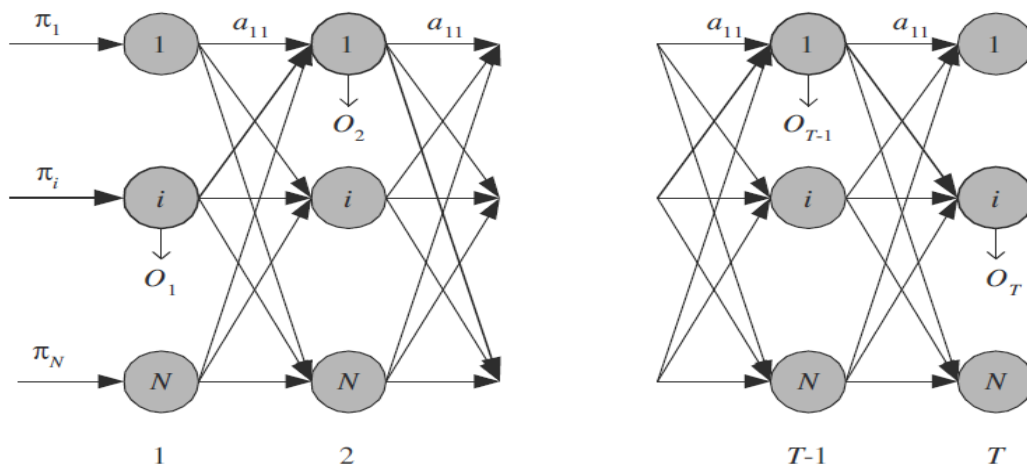


Figure 5 gives an HMM unfolded in time as a lattice (or trellis) showing all the possible trajectories. One path, shown in thicker lines, is the actual (unknown) state trajectory that generated the observation sequence.

Questions that matters in Hidden Markov models:

- i) With a known emissions sequence, determine most likely state path.
- ii) With a known emissions sequence, how can one estimate transition as well as emission probabilities of a model?
- iii) Determine a “forward probability” that a model produces a particular sequence.
- iv) Determine “posterior probability” that a model is in a certain state for any point in a sequence.

#### **4.4 Analysing Hidden Markov Models**

At any point of time we have partial information about any non-trivial situation (limited windows of information). In such cases Probability and Markov process give a way to deal with uncertainty. Hence ultimately: HMM is an elegant tool for this, as it enoporates these two, in such a way we can predict the sequence as we need to trace attack trace in the cloud computing system. In so doing we fulfil the notion of predicting something that is “Hidden” from “Observed”

In the operation and implementation of HMM transition and observation probabilities are combined to give a more compacting machine representation. This is done by combining probabilities; these combined probabilities (referred to as joint event/ joint probabilities) will be combined in the state transitions. More so, Markov assumption is necessary and enables us to draw the state machine.

The main goal in the prediction process is to maximise  $P(S/O)$  where  $S$  is the state sequence and  $O$  is the observation sequence, i.e.  $S^* = \arg \max_s (P(S/O))$ . Markov Assumption and Naïve Bayes is a very powerful tool for problem solving in Statistical AI:

- Markov Assumption: The probability of a state, being the state of the machine, depends only on the previous state: (Called Oder-1 Markov Assumption).
- We can have Oder-k Markov Assumption: Where a state depends on previous  $k$  states

In HMM implementation terms are grouped,

- So that we can capture the start and end of the process
- So the we can combine and introduce the useful notation

The beauty of introducing useful notation is that this turns the probabilistic expression into a probabilistic finite state machine/automaton.

4.4.1 Overall implementation demonstration of HMM in the cloud

	$H_1$	$H_2$	$H_3$
$H_1$	0.1	0.4	0.5
$H_2$	0.6	0.2	0.2
$H_3$	0.3	0.4	0.3

and

	$I$	$V_{2-(T-1)}$	$V_T$
$H_1$	0.3	0.5	0.2
$H_2$	0.1	0.4	0.5
$H_3$	0.6	0.1	0.3

The demonstration shows what happens in the implementation of HMM in the cloud, given an observation sequence:  $I V_2 V_3 V_4 V_T$ , and then tasked to find state sequence that would have produced or resulted in the given observation sequence. The briefing regarding this is that it is not easily computable; hence a real analysis of this is needed.

**Figure 6: Implementation demonstration diagrammatic representation**

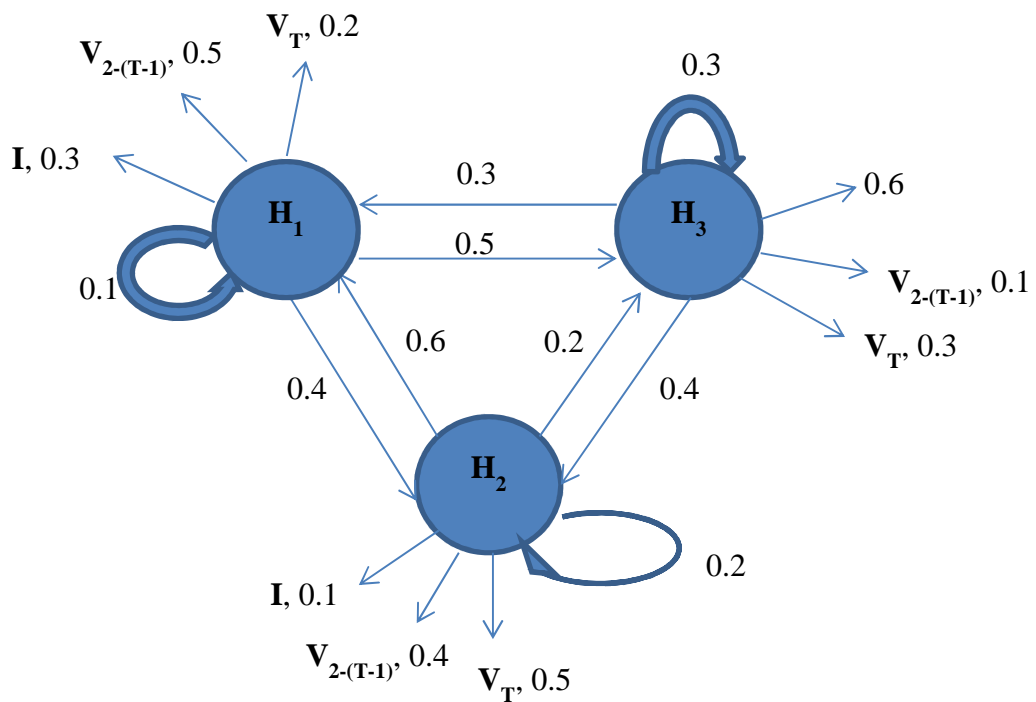
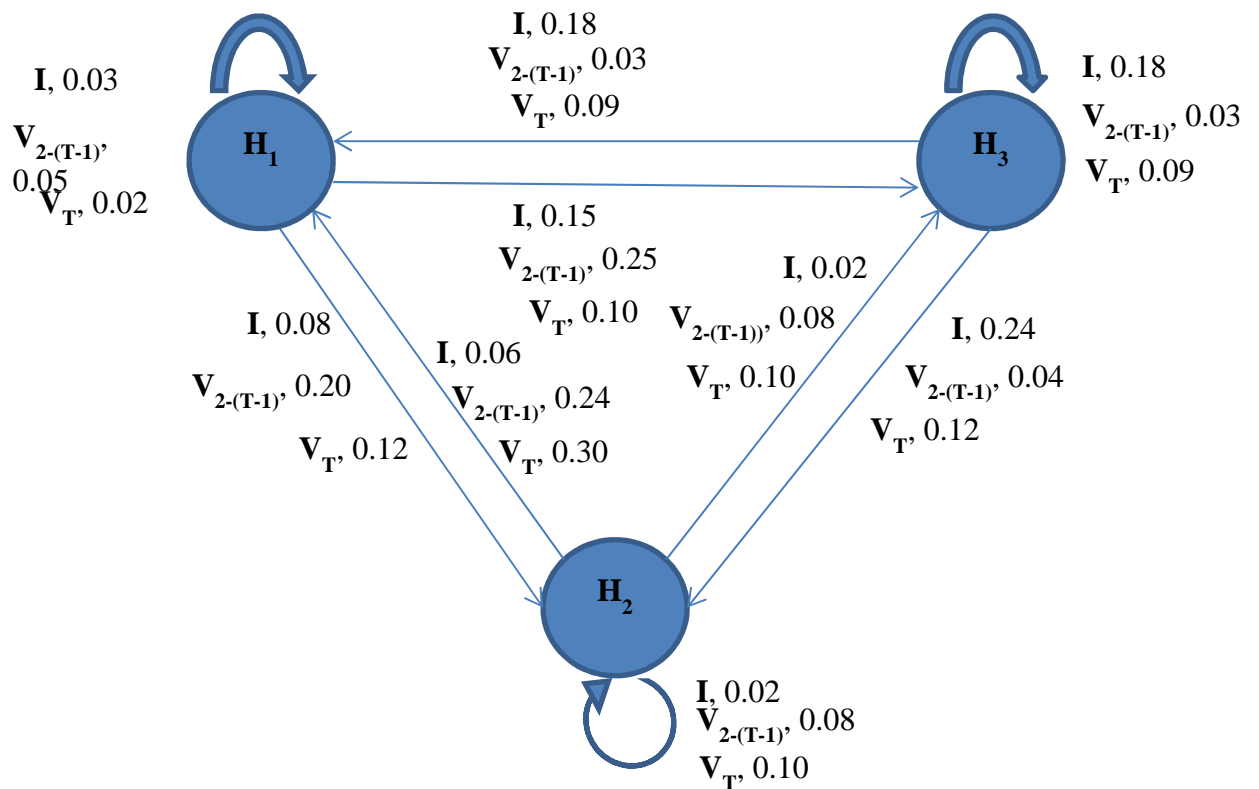




Figure 6 shows the implementation demonstration diagrammatic representation, where the transition probability and observation probability are extracted from the given exemplary tables into a diagrammatic pictorial figure.

**Figure 7: The combined compacted probabilities (Joint event/ Joint probabilities) representation**



As shown in figure 7, the combined probabilities (Joint event/ Joint probabilities) will be combined in the state transitions for a compact representation.

**Figure 8: Viterbi Algorithm for detection problem (first two symbols)**

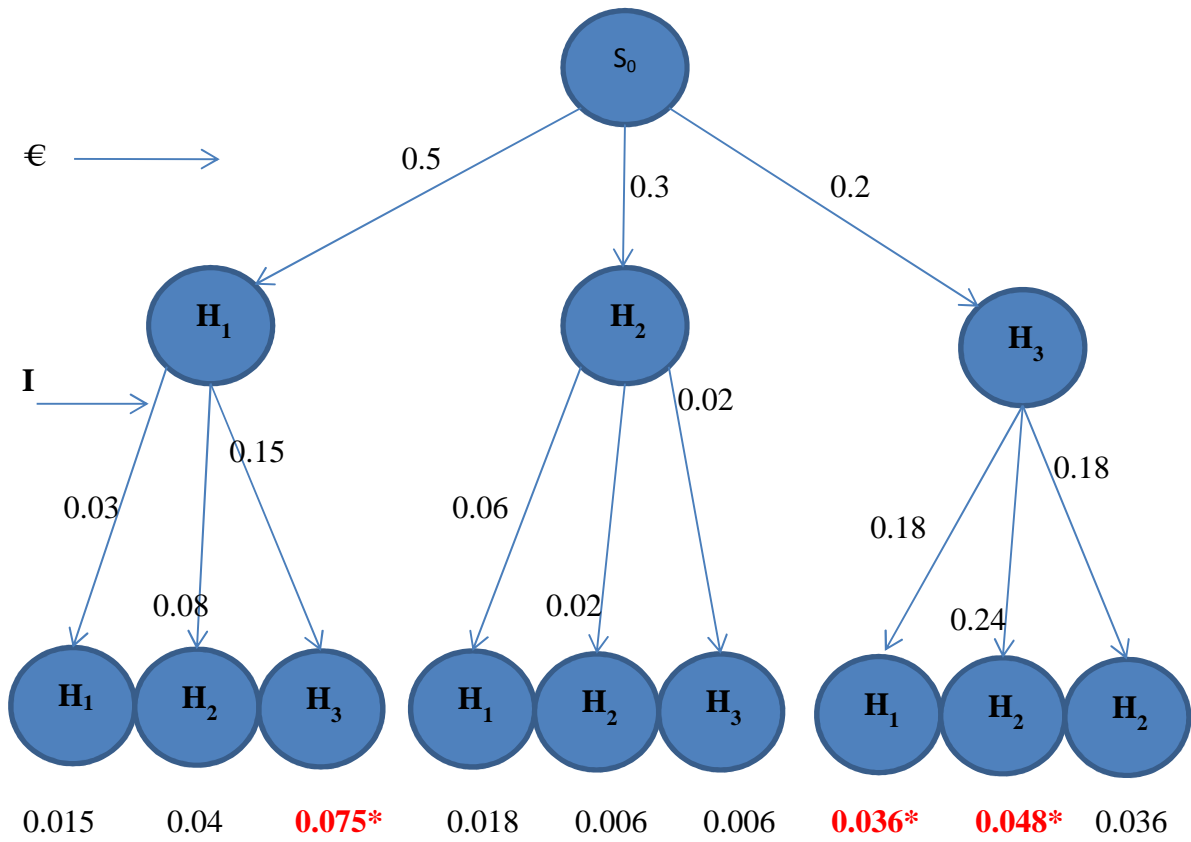


Figure 8 gives an illustration of how Viterbi Algorithm for detection problem is calculated, in this case, only the first two symbols are taken into consideration, for demonstration purposes.

#### 4.4.2 Essentials of Hidden Markov Model

1. Markov + Naïve Bayes
2. Uses both transition and observation probability

$$P(S_k \xrightarrow{O_k} S_{k+1}) = P(O_k/S_k) P(S_{k+1}/S_k)$$

3. Effectively makes Hidden Markov Model a Finite State Machine (FSM) with probability

#### 4.4.3 Properties of the Markov Process

- 1) Limited Horizon: Given previous  $t$  states, a state  $i$ , is independent of preceding 0 to  $t-k+1$  states.

- $P(X_t=i/X_{t-1}, X_{t-2}, \dots, X_0) = P(X_t=i/X_{t-1}, X_{t-2}, \dots, X_{t-k})$
- Order k Markov process

- 2) Time invariance: (shown for k=1)

- $P(X_t=i/X_{t-1}=j) = P(X_1=i/X_0=j) \dots = P(X_n=i/X_{n-1}=j)$

#### 4.4.4 Probabilistic Inference

O: Observed Sequence

S: State Sequence

Given O find  $S^*$  where  $S^* = \arg \max_s (P(S/O))$  called Probabilistic Inference. Inferring “Hidden” from “Observed” is different from logical inference based on propositional or predicate calculus, in some way as shown in the following tabulated table 5.

**Table 5: Probabilistic checked with Logical Inference**

<b>Probabilistic Inference</b>	<b>Logical</b>
Numerical Numbers	Symbolic Expressions
Argument Expression	Inference Rule like Modus ponens
Laws of probability	Laws of Boolean Algebra
Axioms of probability	Axioms of Hilbert, (e.g. in Proposition Calculus)
No concept of soundness or completeness	Soundness, consistency, completeness

The process of marginalisation is used in finding the Probability of Observation Sequence,

with S being used as the margin symbol as follows:

$$\begin{aligned}
 P(O) &= \sum_S P(O, S) \\
 &= \sum_S P(S)P(O/S)
 \end{aligned}$$

Without any restriction, search space size= $|S|^{|O|}$ . This is expressed by means of:

$$\begin{aligned}
 P(O) &= P(O_0).P(O_0/O_1).P(O_2/O_1O_0).P(O_3/O_2O_1O_0).P(O_3/O_2O_1O_0).P(O_4/O_3O_2O_1O_0). \\
 &P(O_k/O_{k-1}O_{k-2}...O_0)
 \end{aligned}$$

In making some deductions (that result in the linear complexity of Viterbi algorithm), very important two probability laws used are:

- 1) Chain Rule

$$P(X_1X_2...X_k) = P(X_1).P(X_2/X_1).P(X_3/X_2X_1)...P(X_k/X_{k-1}X_{k-2}...X_1)$$

- 2) Marginalization

$$P(A) = \sum_{B_1, B_2, \dots, B_N} P(A, B_1, B_2, \dots, B_N)$$

(Done for all possible values of B)

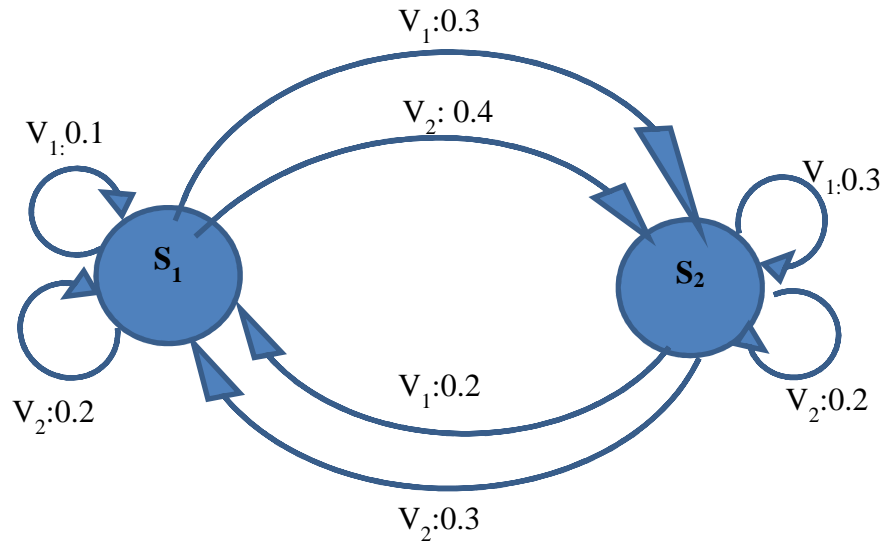
Because of Viterbi complexity down from  $|S|^{|O|}$  to  $|S| \cdot |O|$ , i.e from exponential to linear.

The reason for complexity reduction is that Viterbi is based on dynamic probability of which, Markov Assumption is the key element. Considerations to be take into cognisance when effecting the Viterbi algorithm:

- Transition probability table will have tuples on rows and states on columns
- In the Viterbi, the Markov process will take effect from the 3<sup>rd</sup> input symbol ( $\epsilon$ RR)
- Sequences ending in same tuples will be compared

Hence is doing so as all the considerations are noted, the Viterbi algorithm is used for predicting the state sequence given the observation sequence. It is a very efficient algorithm, with time complexity  $=|S| \cdot |O|$  (i.e. Number of states \* length of observations). During the execution process, every cell records the winning probability ending in that state.

**Figure 9: Probabilistic Finite State Machine (FSM) implementation (Viterbi Algorithm Illustration)**

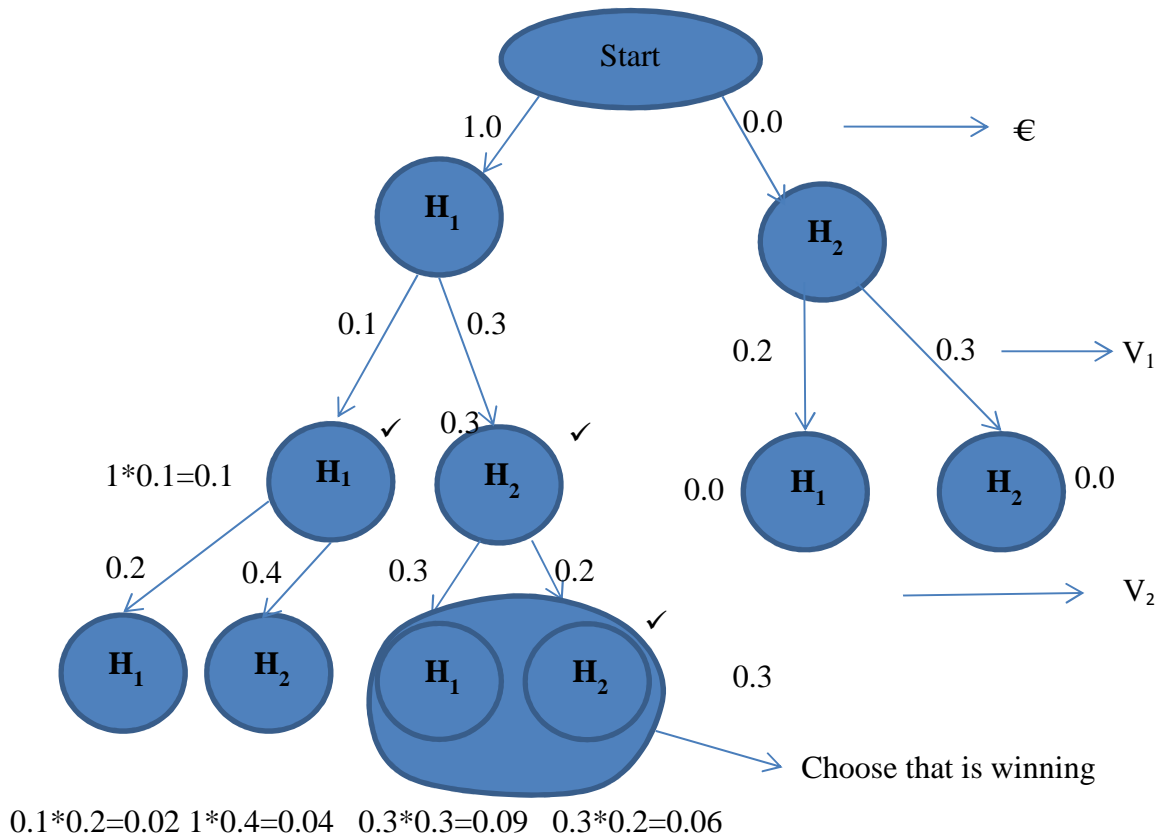


From figure 9, we may have to determine:

$$S^* = \arg \max P(S / V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_2, \mu)$$

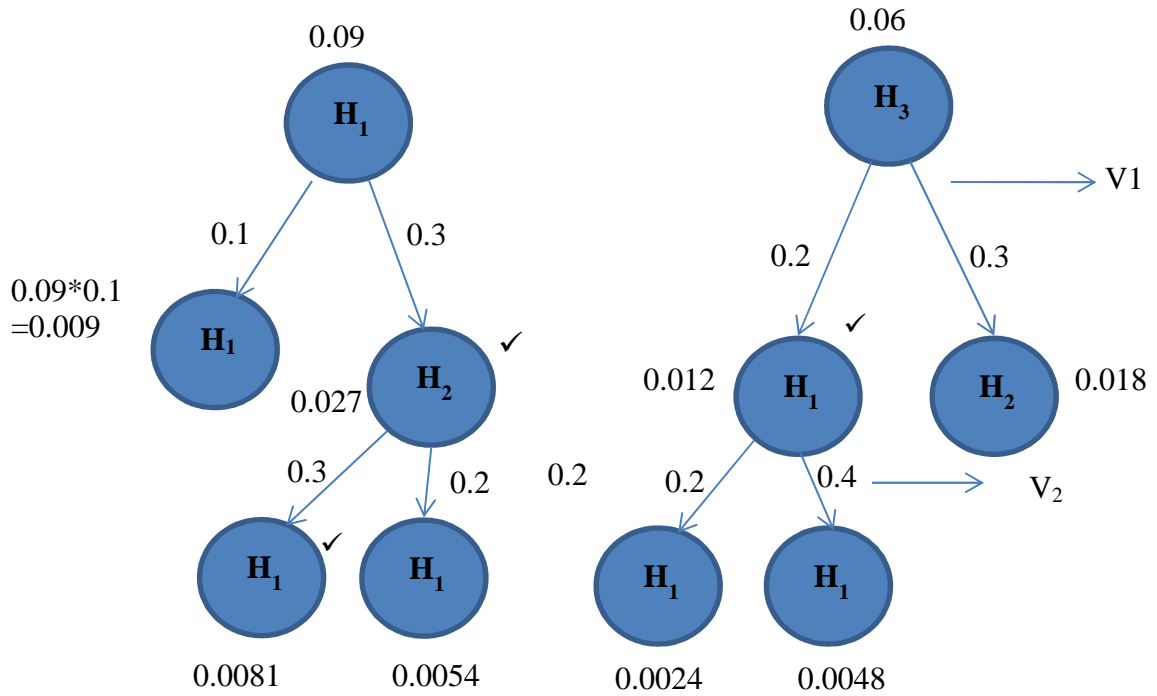
$V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_2$  is the output sequence and  $\mu$  the model or the machine

**Figure 10: Developing the tree of the FSM**



Deriving it from figure 9, figure 10 details on how we can develop the tree of the FSM until we reap the winning nodes. Proceeding with the task, figure 11 shows continued tree development as table 6 tabulates the calculated obtained probabilities giving out the desired path.

**Figure 11: Developing the tree continued....**



**Table 6: Tabular representation of the tree**

Final Winner Ending State	€	$V_1$	$V_2$	$V_1$	$V_2$
$H_1$	1.0	$(1.0 \cdot 0.1, 0.0 \cdot 0.2)$ $= (0.1, 0.0)$	$(0.02, 0.09)$	$(0.009, 0.012)$	$(0.0024, 0.0081)$
$H_2$	0.0	$(1.0 \cdot 0.3, 0.0 \cdot 0.3)$ $= (0.3, 0.0)$	$0.04, 0.06$	$(0.027, 0.018)$	$(0.0048, 0.0054)$

**Note:** Every cell records the winning probability ending in that state

Final winner



#### 4.4.5 Viterbi Algorithm

Given:

- 1) The HMM which means:
  - a) Start states:  $S_1$
  - b) Alphabet:  $A = \{a_1, a_2, a_p\}$
  - c) Set of States:  $S = \{S_1, S_2, \dots, S_n\}_{a_k}$
  - d) Transition probability  $P(S^i \rightarrow S^j)$  For all  $i$  and  $j$   
which is equal to  $P(S_{j,a_k}/S_i)$
- 2) The output string  $a_1, a_2, \dots, a_T$

To find:

The most promising sequence of states  $C_1 C_2 \dots C_T$  which produces a given output sequence, i.e.,  $C_1 C_2 \dots C_T = \arg \max_s [P(C/a_1, a_2, \dots, a_p, \mu)]$

Data Structure is described as follows:

1. A  $N \times T$  array called SEQSCORE to maintain the winner sequence always  
( $N$ =Number of states,  $T$ =Length of o/p sequence)
2. Another  $N \times T$  array called BACKPTR to recover the path.

Three distinct steps in the Viterbi implementation:

- 1) Initialization
- 2) Iteration
- 3) Sequence Identification

1. Initialization

SEQSCORE(1,1)=1.0

BACKPTR(1,1)=0

For (i=2 to N) do

SEQSCORE (i, 1) =0.0

(Expressing the fact that first state is  $S_1$ )

## 2. Iteration

For (i=2 to N) do

For (j=1 to N) do

SEQSCORE (i,t)=Max<sub>(j=1, N)</sub>  
[SEQSCORE (j, (t-1))\*P ( $S_j \xrightarrow{a_k} S_i$ )]

BACKPTR (I,t)=index  $j$  that gives the MAX above

## 3. Sequence Identification

C(T)=i that maximises SEQSCORE(i, T)

For i from (T-1) to 1 do

C(i)=BACKPTR[C(i+1),(i+1)]

Optimizations possible are:

1. BACKPTR can be  $1 * T$
2. SEQSCORE can be  $T * 2$

#### 4.4.6 Forward $F(k,i)$ and Backward $B(k,i)$ Probability Calculation

Forward Probability  $F(k,i)$ :

- Define  $F(k, i)$  = Probability of being in state  $S_i$  having seen  $O_0O_1O_2...O_k$
- $F(k,i) = P(O_0O_1O_2...O_k, S_i)$
- With  $m$  as the length of the observed sequence
- $P(\text{Observed sequence}) = P(O_0O_1O_2...O_m)$

$$= \sum_{p=0, N} P(O_0O_1O_2...S_p)$$

$$= \sum_{p=0, N} P(m, p)$$

$F(k,q)$

$$= P(O_0O_1O_2...O_k, S_q)$$

$$= P(O_0O_1O_2...O_k, S_q)$$

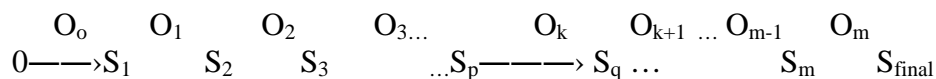
$$= P(O_0O_1O_2...O_{k-1}, O_k, S_q)$$

$$= \sum_{p=0, N} P(O_0O_1O_2...O_{k-1}, S_q, O_k, S_q)$$

$$= \sum_{p=0, N} P(O_0O_1O_2...O_{k-1}, S_p)$$

$$P(O_m, S_p / O_0O_1O_2...O_{k-1}, S_p)$$

$$= \sum_{p=0, N} F(k-1, p) \cdot P(S_p \xrightarrow{O_k} S_q)$$



The Boundary condition for Forward Probability:  $F(0,q) = P_q$ , where  $P_q$  is the initial probability of being in state  $S_q$ , i.e.  $(S_0 \rightarrow S_q)$ . Forward probability can be computed in time proportional to the stretch of observation sequence. So it is a linear time computation.

Hence the complexity of the forward probability calculation is  $|S| \cdot |O|$ : (Where  $|S|$  equals Number of states;  $|O|$  Observation sequence's stretch)

Backward Probability  $B(k,i)$  Calculation:

- Define  $B(k,i)$  = Probability of  $O_k O_{k+1} O_{k+2} \dots O_m$ , given that the state was  $S_i$
- $B(k,i) = P(O_k O_{k+1} O_{k+2} \dots O_m / S_i)$
- With  $m$  as the length of the observed sequence
- $P(\text{Observed sequence}) = P(O_0 O_1 O_2 \dots O_m)$   
 $= P(O_0 O_1 O_2 \dots O_m / S_0)$   
 $= B(0,0)$

$B(k,q)$

$$= P(O_k O_{k+1} O_{k+2} \dots O_m / S_q)$$

$$= P(O_{k+1} O_{k+2} \dots O_m, O_k / S_q)$$

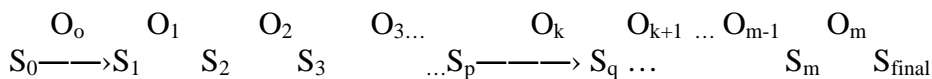
$$= \sum_{q=0, N} P(O_{k+1} O_{k+2} \dots O_m, O_k, S_q / S_p)$$

$$= \sum_{q=0, N} P(O_k, S_q / S_p)$$

$$P(O_{k+1} O_{k+2} \dots O_m / O_k, S_q, S_p)$$

$$= \sum_{q=0, N} P(O_{k+1} O_{k+2} \dots O_m / S_q) \cdot P(O_k, S_q / S_p)$$

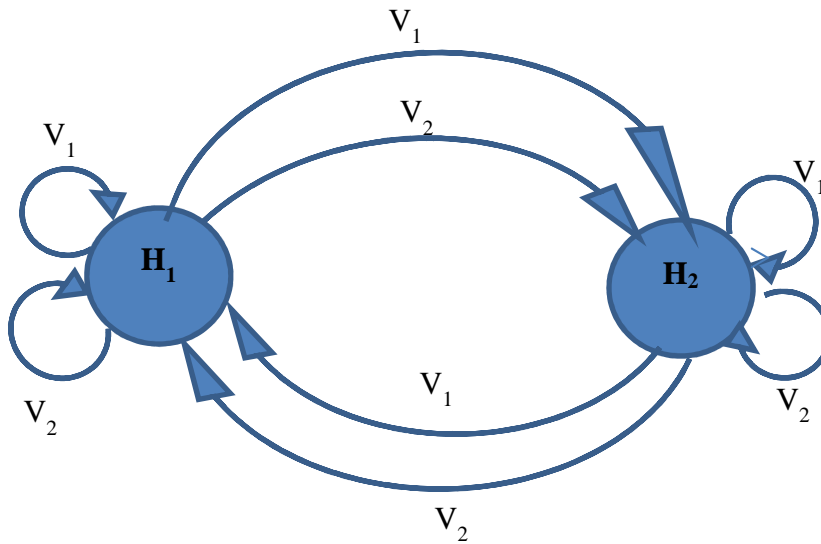
$$= \sum_{q=0, N} B(k+1, q) \cdot P(S_p \xrightarrow{O_k} S_q)$$



The boundary condition for the backward probability:  $B(K, P)$  is obtained from the last symbol  $B(m, \text{final})$  where  $S_{\text{final}}$  is one of the state of the HMM. Both Forward and Backward Probability have a linear time complexity and have a recursive nature

#### 4.4.7 HMM Training (Baum-Welch Algorithm) Counts

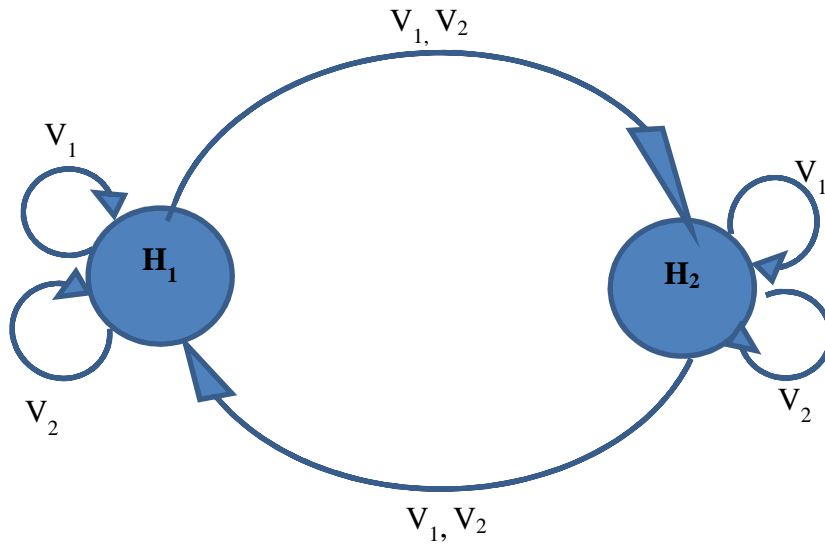
**Figure 12: Key Intuition**



Given:	Training sequence
Initialisation:	Probability values
Compute:	$P(\text{state seq/ training seq})$ , get count of transition, compute rule probabilities
Approach:	Initialise the probabilities and recomputed them

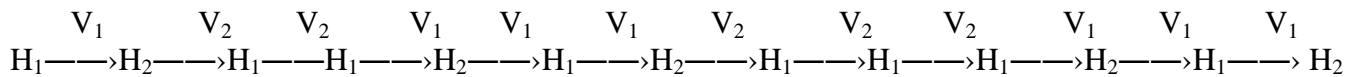
Figure 12 gives the key intuition of the HMM training (Baum-Welch Algorithm) counts, with the labels that shows what happens. Beneath are given details that need consideration when making use of Baum-Welch Algorithm. Follows is figure 13 together with table 7 that contains some counts, we demonstrate on how calculations are delivered for a given exemplary string.

**Figure 13: Baum-Welch Algorithm: Counts**



String =  $V_1V_2V_2 V_1V_1V_1 V_2V_2V_2 V_1V_1V_1$

Sequence of states with respect to input symbols



Calculating probabilities from table:

**Table 7: Table of counts**

Src	Dest	O/P	Count
$H_1$	$H_2$	$V_1$	5
$H_1$	$H_1$	$V_2$	3
$H_2$	$H_1$	$V_1$	3
$H_2$	$H_1$	$V_2$	2

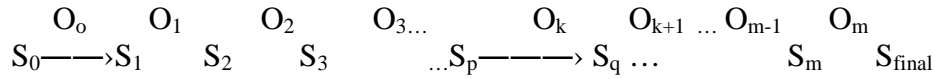
$$P(H_1 \xrightarrow{V_1} H_2) = 5/8$$

$$P(H_1 \xrightarrow{V_2} H_1) = 3/8$$

$$P(S^i \xrightarrow{W_k} S^j) = \frac{C(S^i \xrightarrow{W_k} S^j)}{T \sum_{m=1}^A W_m} = \sum_{t=1}^T \sum_{m=1}^A C(S^i \xrightarrow{W_k} S^j)$$

T=Number of states; A=Number of Alphabet symbols

Now if we have a non-deterministic transitions then multiple state seq possible for the given o/p seq. Our aim is to find expected counted through this.



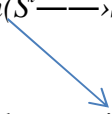
$C(S^i \xrightarrow{W_k} S^j)$  is formed weighting the number of appearances by  $P(S/W)$ .

4.4.8 Interplay between Two Equations (Expectation Maximisation)

$$P(S^i \xrightarrow{W_k} S^j) = \frac{C(S^i \xrightarrow{W_k} S^j)}{T \cdot A \cdot W_m}$$

$$= \sum_{t=1} \sum_{m=1} C(S^i \xrightarrow{W_k} S^j)$$

$$C(S^i \xrightarrow{W_k} S^j) = \sum_{S_{0,n+1}} [P(S_{0,n+1}/W_{0,n}) * n(S^i \xrightarrow{W_k} S^j, S_{0,n+1}, W_{0,n})]$$


 {No. of time the transitions  $S^i \xrightarrow{W_k} S^j$  occurs in the string}

Baum welch learns probability values on arcs. It is important to note that the structure of HMM is not important. If multiple observation sequences are given, get the new probabilities for each observation sequence. Do this for one iteration. Get the values after one iteration over all observations. This is called an epoch, which a way of running the Baum Welch algorithm. So a way of defining an epoch: There has to be one iteration over all observations patterns, and after each iteration we get the updated probabilities. This is done until convergence.

#### 4.4.9 Computational Part

$$\begin{aligned}
& C(S^i \xrightarrow{W_k} S^j) \sum_{S_{0,n+1}} [P(S_{0,n+1}/W_{0,n}) * n(S^i \xrightarrow{W_k} S^j, S_{0,n+1}, W_{0,n})] \\
&= \frac{1}{P(W_{0,n})} \sum_{S_{0,n+1}} [P(S_{0,n+1}, W_{0,n}) * n(S^i \xrightarrow{W_k} S^j, S_{0,n+1}, W_{0,n})] \\
&= \frac{1}{P(W_{0,n})} \sum_{t=0, n} \sum_{S_{0,n+1}} [P(S_t=S^i, W_t=W_k, S_{t+1}=S^j, S_{0,n+1}=S^j, W_{0,n})] \\
&= \frac{1}{P(W_{0,n})} \sum_{t=0, n} [P(S_t=S^i, W_t=W_k, S_{t+1}=S^j, W_{0,n})] \\
&= \sum_{t=0}^n P(S_t=S^i, S_{t+1}=S^j, W_t=W_k, W_{0,n}) \\
&= \sum_{t=0}^n P(W_{0, t-1, n} | S_t=S^i, S_{t+1}=S^j, W_t=W_k, W_{t+1, n}) \\
&= \sum_{t=0}^n P(W_{0, t-1, n} | S_t=S^i) P(S_{t+1}=S^j, W_t=W_k | W_{0, t-1}, S_t=S^i) P(W_{t+1, n} | S_{t+1}=S^j) \\
&= \sum_{t=0}^n F(t-1, i) P(S_{t+1}=S^j, W_t=W_k | S_t=S^i) B(t+1, j) \\
&= \sum_{t=0}^n F(t-1, i) P(S_{t+1}=S^j \xrightarrow{W_k} S^i) B(t+1, j) \\
& \quad \begin{array}{cccccccc} W_0 & W_1 & W_2 & W_k & W_{n-1} & W_n \\ S_0 \xrightarrow{\quad} & S_1 \xrightarrow{\quad} & S_2 \xrightarrow{\quad} & \dots & S_j \xrightarrow{\quad} & S_{n-1} \xrightarrow{\quad} & S_n \xrightarrow{\quad} & S_{n+1} \end{array} \\
& P(W_{0,n}) = \sum_{t=0}^n F(t-1, i) \cdot B(t+1, i) \\
& S^i \xrightarrow{W_k} S^j
\end{aligned}$$



#### **4.5 Description of the proposed framework**

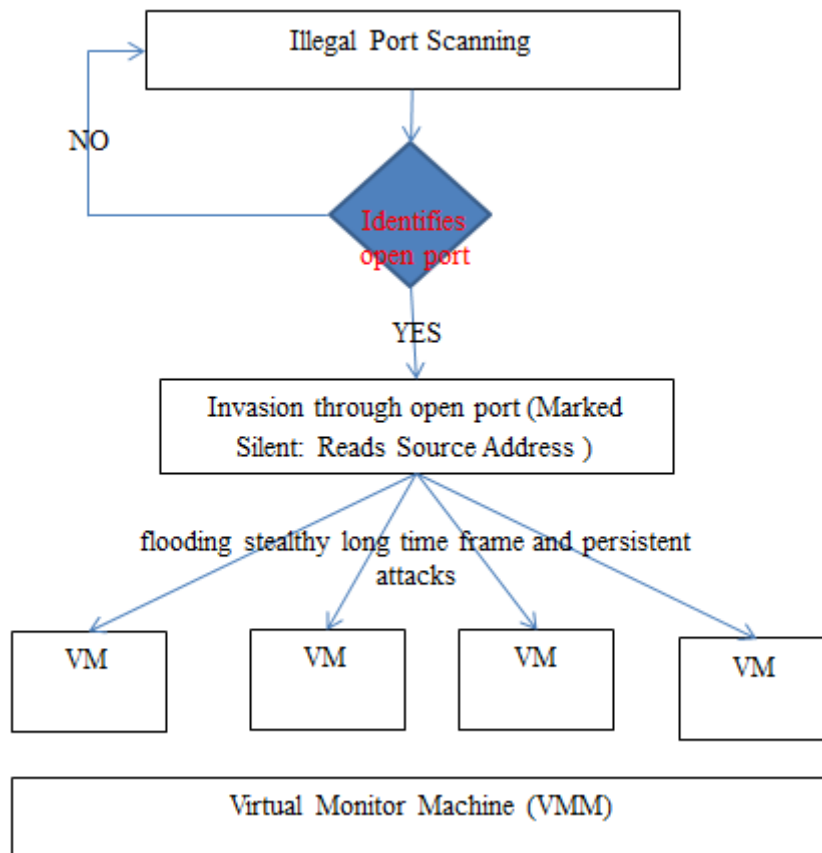
The proposed framework uses Hidden Markov Model which is a mathematical statistical model that incorporates three different algorithms to accomplish our desired goal. We bank on this model because of its capability to model hidden parameters and sequences. It has found its good use in the field of Natural Language Process and we need to transpolate this into cloud computing as we detect the DDoS attack in the cloud, centred on the flooding that takes place in Virtual Machine (VM).

The proposed cloud DDoS trace system analyses a number of logs, thus the recorded in logs that were intended to flooding stealthy, long time frame and persistent attacks to targeted virtual machines. By the definition of the silent/stealthy attacks, which in this instance are being considered to invade the system through open port following the illegal port scanning activity, hence the system trace these attacks to have invaded the cloud computing system through the previous undesignated ports in the cloud computing system.

##### *4.5.1 Resembling the attacker scenario*

DDoS attack can make cloud service unavailable, it starts from the port scanning process that is done by the attacker around the cloud computing system is considered as illegal, since the intentions are to find the weakness point onto which the cloud computing system can be invaded. In this silent attack, the attacker identifies open ports, once the open port/s are identified, the attack invade the cloud using that port and flood some persistent silent long time frame requests to the targeted victim Virtual Machine (VM), by so doing the invasion will be labelled as silent, since at this moment it does not trigger any alert triggers, and figure 14 shows the pictorial view of this scenario.

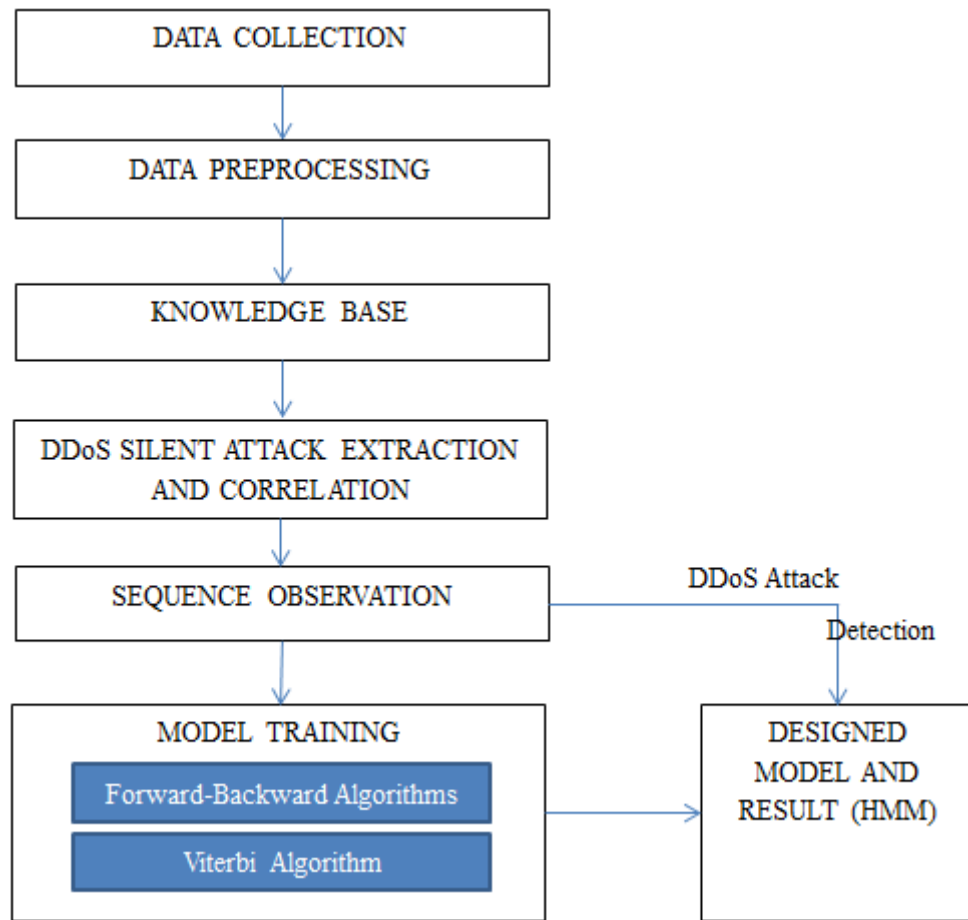
**Figure 14: Resembling the attacker scenario**



*4.5.2 Design, Modelling and Experimentation of the DDoS Silent attack detection framework*

The main goal is to monitor VM cloud requests activities, determining their source IP addresses and the activities that they would have performed in the cloud system. High accuracy, high detection rate and lower down the false alarm rate have to be maintained, (i.e. flooding attack trace/s). After designing the DDoS cloud detection system/model, with the help of collected testing data, we evaluate the performance of designed model, though an implementation and simulation analysis.

**Figure 15: Proposed Framework**



The proposed framework as shown in figure 15 is described as follows:

**Data Collection:** This is the collected data from especially audit logs. In this thesis work we first generated our data as it suites us for our intended simulation purposes.

**Data pre-processing:** It is the critical and crucial step that results in the final training set. If professionally done, the resulting training set that will be kept in the knowledge base will be of good quality.

Knowledge base (KB) in our proposed model stores and is a depository for resulting training set in relation to our intended goal and purpose of DDoS attack sequence detection.

DDoS Silent Attack Extraction and Correlation: The analysed logs which have tracked to have reached their request to the Virtual Machine through open ports are taken into cognisance, (this assumption can be as rule in detecting process), hence summarisation of trace of requests rooted to the virtual machine. This helps in determining their persistence in terms of time frames, and if it is identified to be the flooding the requests, we can trace the IP address in which it has invaded (get access into) the system, in this case obviously previously unassigned ports (i.e. undesignated IP addresses) which will qualify to be silent DDoS attacks.

Sequence of observations: This is a number of numerous observation sequences that are linked to the hidden state sequence, more so it can happen that the obtained observations may not be coordinated to each other. In turn they aid HMM to detect attack sequences in cloud; they model the attacker's intentions.

Model Training (Forward-Backward algorithm and Viterbi algorithm): Forward-Backward algorithm has a special name called Baum-Welch algorithm finds the optimal state transition probabilities and observation sequence probabilities, if given a set of observations. Baum-Welch learns probability values on arcs: If multiple observation sequences are given, for one iteration, get the new probabilities for each observation sequence. Get the values after one iteration over all observations. This is called an epoch, which is a way of running the Baum Welch algorithm. So a way of defining an epoch: There has to be one iteration over all observations patterns, and after each iteration we get the updated probabilities. This is done until convergence.

The Viterbi algorithm gives the most-probable (Viterbi) path through the HMM state, out of pool of given observations along with HMM. In Viterbi, the Markov process will take effect from the 3<sup>rd</sup> input symbol ( $\epsilon RR$ ). The Viterbi algorithm has three (3) distinct steps in the Viterbi implementation, which are Initialization, Iteration and Sequence Identification.

DDoS Silent Attack Detection: Extracted silent invasions with their observed trends are assembled into the perceived activities, followed by being fed in a trained HMM for analysis thus give out a result, if DDoS silent attack sequence has been detected or the other way round. The trend (in matrix inform) which will result from new captured rooted request is compared with already stored knowledge representing normal client/consumer request/s. After detecting the flooding DDoS attacks then the attacker source IP address has to be identified too, this is achieved by counting the value of TTL (Time\_to\_Life) [40] which gives the number of nodes the attacker passed through until it reaches victim side, starting from the vulnerable node/s.

Designed Model and result (HMM): This is the prototypical with specific or set HMM strictures and evaluates its performance.

#### **4.6 Performance Evaluation Measures**

Our evaluation is based on the successfulness of our model in detecting an attack sequence given that the silent invasion has taken place and the virtual machine have been flooded. We have to make a determination of a series of the flooded VMs, from the silent scanned port, to the vulnerable VM, through a sequence of invaded machines to the last targeted VM which the hacker or attacker might be interest in. In the intended simulation, we have to a have numbered sequence of certain observations which we will randomly select and check on its parameters, that have the highest chances of produces that sequence, by doing

so, we will be determining it's source of origin in a way. If the new set of data is given into the system, the system must be able to re-generate new set of observation sequence/s where we can again depict the most probable state sequence that would resulted in the attack sequence (through a series of VMs from vulnerable- to-a chain of flooded-VMs-to the final target VM) produced or obtained. An attack sequence is derived from the states and transitions. In our case the states are a group of attackers (attacker level, resembling different interests of attackers). The possible transitions are port scanning (P), vulnerability (V) and DDoS attacking by multiple flooding requests (A).

## CHAPTER 5

### Result Analysis

In this chapter, we have shown and explained all the results obtained in this research. The univariate and multivariate results are described. Pictorial combined textual words are used for the easiness of the interpretation of the results graphically and in words. The results of our proposal (Interpolating properties of HMM into the field of Cloud Computing) are shown in this unit.

#### 5.1 The estimate of a statistical model according to a training set

The probabilistic sense contributes to the identification of optimal model, which will in turn gives the definite model along with the authentic hidden state sequence values since by merely knowledge the observation sequence is not sufficient to tell model. Searching for the most probable model is the natural and considerably used principle in Markov modelling. In an ideal and simplest case the model is derived from knowledge about the entities we study and its configuration. However it tends out that the knowledge availability is insufficient in voluminous real life situations. Nevertheless, the knowledge can be improved in relation to sourced and dished training data, bringing out the concept termed as learning.

There are two main types of learning, as given in table 8:

- i) The supervised learning: It has  $(x_i, y_i)$  pairs as a training set
- ii) The unsupervised learning: It has  $y_i$  (single attribute) as a training set

Where

- a)  $x_i$  is the corresponding states
- b)  $y_i$  is sequences of observations

**Table 9: Learning comparatives**

<b>1)Supervised learning</b>			<b>(<math>x_i, y_i</math>)</b>	
(i)	Training set	<b>(<math>x_i</math>)</b>		<b>(<math>y_i</math>)</b>
		corresponding states		sequence of observations
(ii)	ML estimation	Pure counting of relative frequency of occurrence of events		
<b>(ii) Unsupervised learning</b>				
(i)	Training set			<b>(<math>y_i</math>)</b>
(ii)	ML estimation			Baum-Welch re-estimation algorithm and is a distinct instance of the EM algorithm (Expectation Maximization), plus it's repetitive in nature so as to suite training data



In our thesis experimental work, we adopted unsupervised learning. The source of the training data and how it may influence the choice of the proper learning algorithm is a great question to consider. The maximum likelihood (ML) estimate is suitable if the training data are random samples from a probability distribution that can be searched for.

Observables/ Observations are the manifestation, vivid to the beholder giving an indication of what would have happened in the hidden/unseen parts of the system. Hidden states are the internally coded information with its effects shown in the observations. Either the random or dominant diagonal method is chosen to initial values.

## **5.2 New training data sampling, sequence probability and probable states path**

- 1) Sampling of the new training data
  - i) The new data samples can be produced from our Markov model by running through the Markov model in a probabilistic way.
  - ii) Data sequence's graphical view can also be provided.

The probability of a given sequence equals probability of sequence having taken from such a prototypical, and is an important problem that needs attention. Problems to be solved in the HMM process

- i) The simulation/experimentation gives an illustration of sequence evaluation (i.e. forward/backward procedure.)
- ii) Summation up the probabilities during the process

The way to calculate the possibility and/or chances that a sequence have been taken from supplied model, is by using use starting state vector  $w$ , the state transition matrix  $WS$ , including emission prospects  $WE$ , from this we calculate the possible state path.

2) The sequence of most probable values of the states: The Viterbi algorithm undertakes this task. It uses the concept of dynamic programming and can be taken to mean as a search for the shortest path.

If we are given a sequence of observations (i.e. an attack trend/attack trace), we work out the perceived symbol sequence, ("Observation sequence), (i.e. an attack trend/attack trace), acquiring states (attacker levels), then electing a path that has the outstanding probability. Sequences of states are hackers/attackers and the emitted symbol sequences are the attack trend/attack traces which are resulting series of virtual machines used, till the flooding was finally done.

### **5.3 Discussion of Results**

The diagrams demonstrations instances of where the hacker moves in its attacking process and leaves an attack trace. This emanates from the consistently silent port scanning, then invading/intruding a vulnerable VM and at last flooding multiple requests to virtual machines resulting in DDoS attack.

Attack numbers are in a way linked to the observables, noticeable trends that take place in the cloud environment. Observables are automatically determined or generated from the given information of the silent attack number. This is based on some given rule in the simulation cloud environment as per our consideration. Observables define the number of

possible number of symbols (VMs, numbering from  $V_1$  to  $V_T$  however these are represented in numbers) that can be found in a sequence.

**Table 10: Simulation Possibilities defined by Silent attack number along with Observables**

Silent attack number	Observables
1	2
2	4
3	8

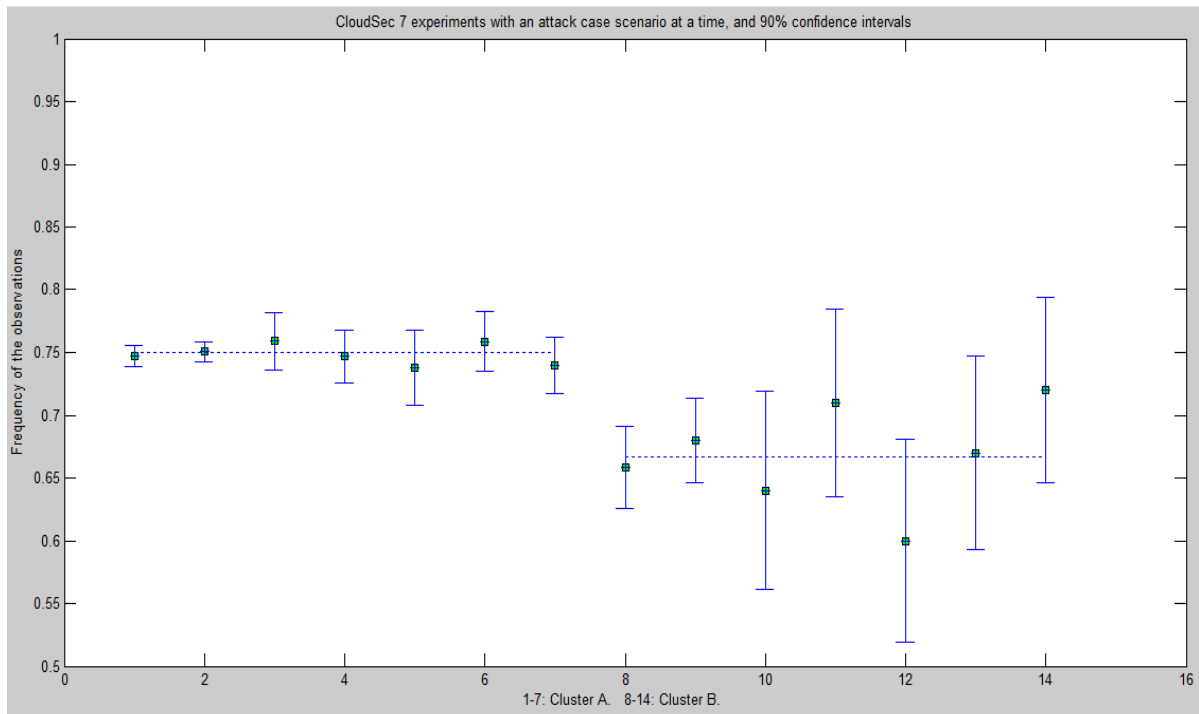
The origins of the used simulation data, is iterated based on the picked simulation/experimentation number. The number of total available sequences are obtained based on from which the simulation/experimentation number is picked or selected, and this is tabulated in table 9.

**Table 11: Available simulation/experimentation number against the total number of sequences**

simulation/experimentation number	Number of sequences
Exp. 1	756
Exp. 2	691
Exp. 3	132
Exp. 4	134
Exp. 5	136
Exp. 6	132
Exp. 7	135

As shown in table 10, the number of sequences is part of component that defines sample size, which is given by the number sequence along with each sequence length. From this, a sequence number can be entered to view the traces that comes in three forms, which are the used prior, transition and observation matrices.

**Figure 16: A graphical representation of the selected data obtained from the selected simulation/experimentation number**



In figure 16, cluster A shows the silence and persistence in the attack scenario, shown by the steadiness of the represented line without any variations. The graphical view shows a resemblance of the intended modelled persistent silent attacks and the severance.

### 5.3. 1 Model estimation from the selected training set.

**Figure 17: Initialisation of the prior, transition and observation matrix as given from the selected simulation number**

```
prior1 =  
  
    0    1    0  
  
transmat1 =  
  
    0.6700    0.1650    0.1650  
    0.1650    0.6700    0.1650  
    0.1650    0.1650    0.6700  
  
obsmat1 =  
  
    0.6700    0.3300  
    0.6700    0.3300  
    0.3300    0.6700
```

Figure 17 gives the initialised prior, transition and observation matrix as given from the selected simulation number, and figure 18 pops the set 15 iterations along with their corresponding estimated log like-hood.

**Figure 18: Iterations with their corresponding estimated log like-hood**

```
loglik_init =  
  
-1.2461e+003  
  
iteration 1, loglik = -1246.142563  
iteration 2, loglik = -982.329157  
iteration 3, loglik = -850.462057  
iteration 4, loglik = -765.866638  
iteration 5, loglik = -726.422462  
iteration 6, loglik = -707.835007  
iteration 7, loglik = -697.956272  
iteration 8, loglik = -692.257557  
iteration 9, loglik = -688.856696  
iteration 10, loglik = -686.816269  
iteration 11, loglik = -685.583154  
iteration 12, loglik = -684.817382  
iteration 13, loglik = -684.317838  
iteration 14, loglik = -683.970244  
iteration 15, loglik = -683.711027
```

**Figure 19: Obtained results after Iterations**

```
loglik =  
  
-683.5049  
  
transmatEM =  
  
    0.7905    0.1868    0.0227  
    0.1087    0.6638    0.2275  
    0.0000    0.0000    1.0000  
  
obsmatEM =  
  
    0.9998    0.0002  
    1.0000    0.0000  
    0.0000    1.0000
```

Figure 19 gives the obtained transitionEM and observationEM matrix after a significant number of iterations.

**Figure 20: The obtained “Expectation Maximisation” training graph, drawn from the first simulation/ experimentation number**

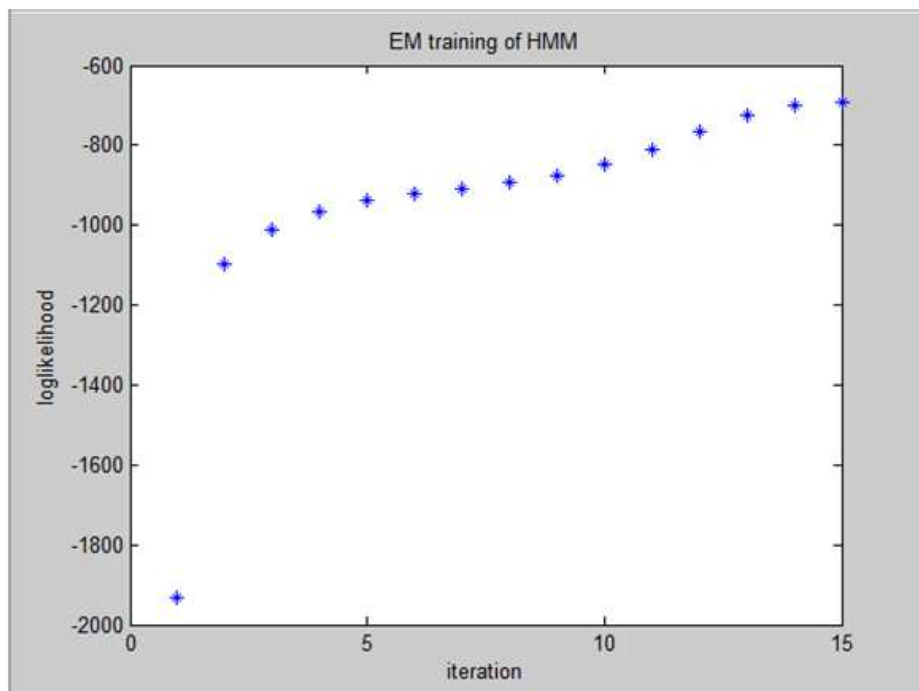


Figure 20 shows whether there is an incremental correlation of the Expectation Maximisation training, towards attaining our goal as number of iterations increases.

**Figure 21: The obtained best parameter view from the simulation**

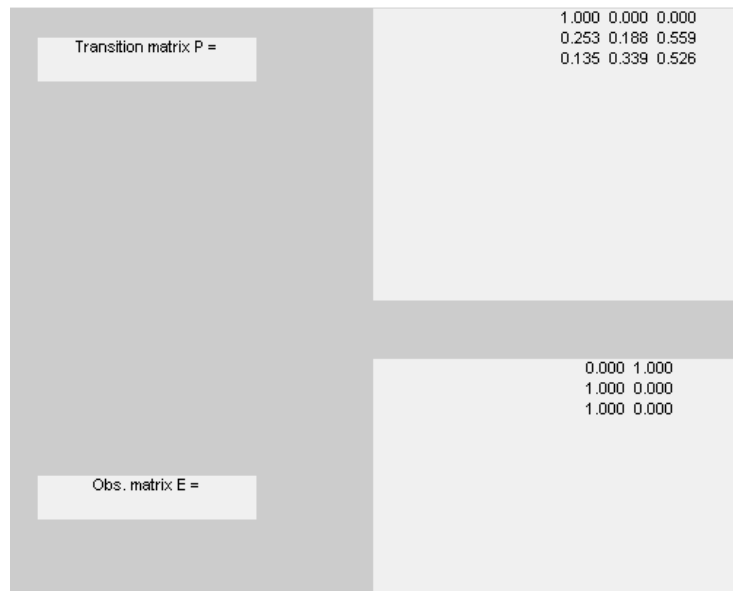
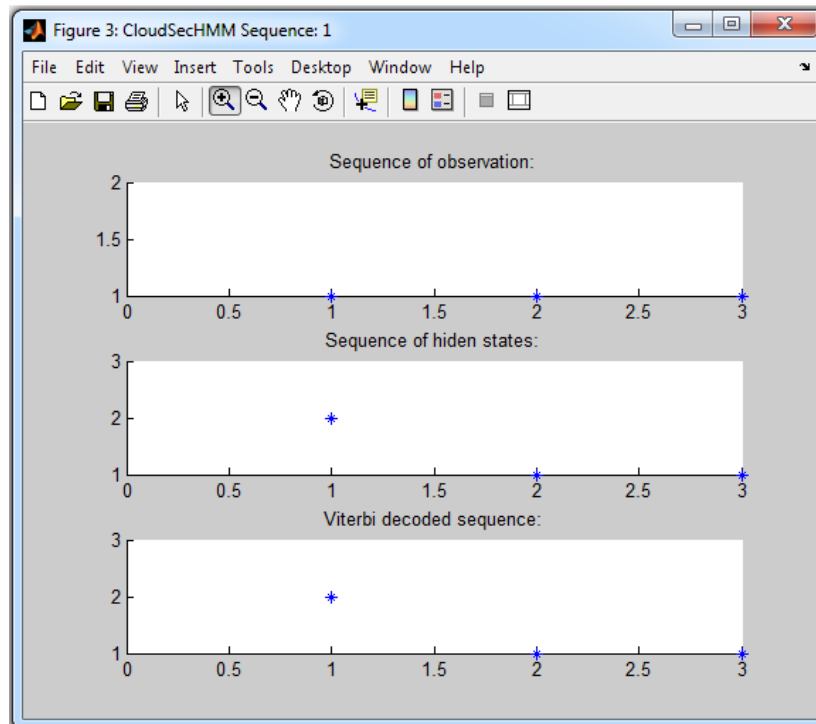


Figure 21 retains the obtained best parameter of the Transition and Observation matrices.

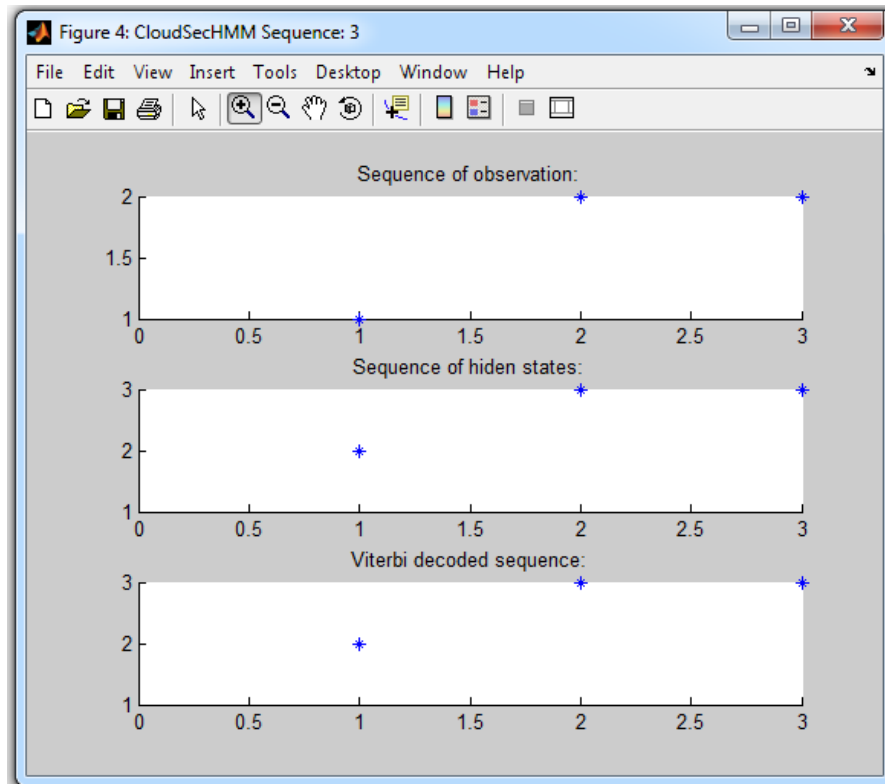
**Figure 22: Sequence 1 along with its hidden states, given a hackers trend used**



The decoded path is 2, 1, 1

Figure 22 gives graphical attack trace that has been decoded, for a selected observation sequence; in this case, it is sequence number 1. Figure 23 and 24 show the same, just a way of depicting the variation of different sequence numbers.

**Figure 23: Sequence 3 along with its hidden states, given a hackers trend used**

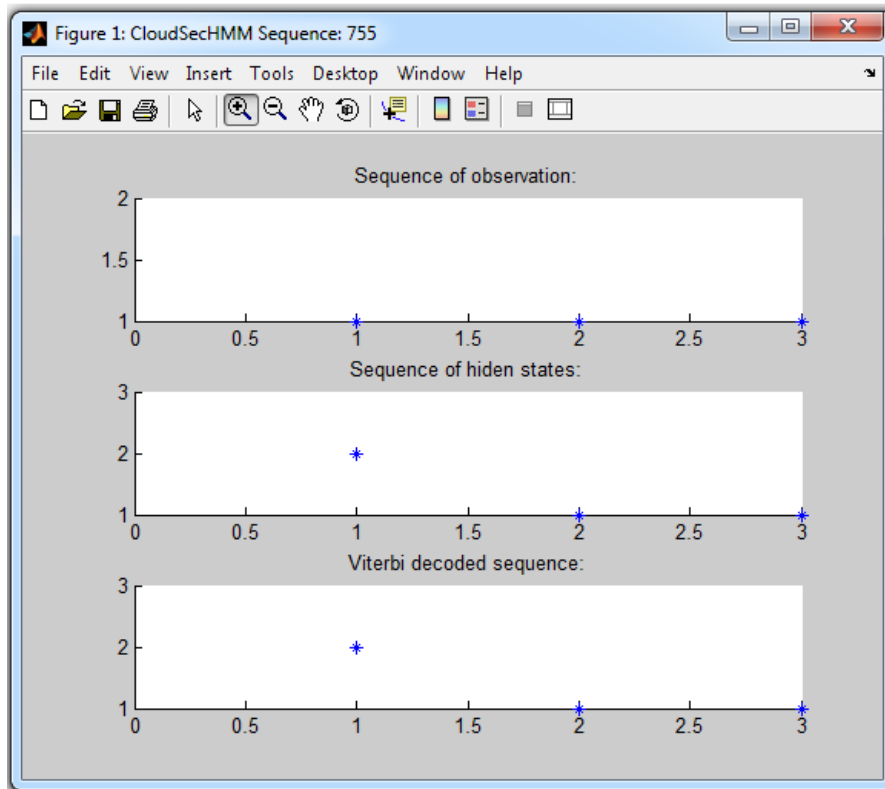


decodedPath =

2 3 3



**Figure 24: Sequence 755 along with its hidden states, given a hackers trend used**



## **CHAPTER 6**

### **Conclusion and Future Work**

This chapter presents conclusion and scope of future work in this area of research. The conclusion summarizes the results of the work conducted in this thesis and describes the future directions in this area of research. The DDoS silent attacks can be detected by identifying hacker-attack sequences: (i.e. attack plan, by way of traffic monitoring). This attack plan phases have been modelled by Hidden Markov model consequently resulting in the successfulness of giving out attack trends of the hacker in terms traces. The proposed system adopts the concept of state to describe the suspicious events in a sequence of states.

#### **6.1 Summary of the thesis**

In **Chapter 1**, we have introduced the topic of our research. It explains the basic concepts of the work done and gives the overview of the thesis. It explains the various concepts related to security paradigms in cloud computing, (i.e. cloud computing as a field itself but in the security lined perspective). This is followed by motivation of the work and goals and objectives of the thesis. The main motivation is the need to increase the level security in the Cloud Computing environment. Cloud security can be increased by taking cognisance and incorporating sound security in contemporary security dimensions and attributes at every cloud service model level. Thus, we have interpolated Hidden Markov Model (HMM) into the Cloud Computing environment, as a way of offering cloud security, tackling the problem of DDoS attack in the virtual machines. There are sequenced objectives which are to detect the silent attacks targeted at virtual machines; giving out its attack trace in turn this enhances cloud security.

After giving the brief introduction in chapter 1, in **Chapter 2** we have reviewed the work already done in this area. It shows the summary of the work done by different researchers laying the main emphasis on the security techniques used, description given and the resulted impacts targeted. For this, we have reviewed various previous studies which have security mechanisms that are related and have attempts to offering security in the cloud. Our review showed that inherited most used cloud computing security mechanism are fine-tuned (tailored) commonly used security techniques previously used in other different domains. We observed that machine learning models and methods/algorithms have potentials to be adopted in this domain.

In **Chapter 3** we have described the background of this research work, the measures and metrics, the cloud security framework metrics along with their definitions and extended descriptions. We also pronounced the difficulty in asserting the independence and dependence of variables used in the study plus that of the empirical data collection are presented in this chapter.

**Chapter 4** has emphasized on the key research concepts i.e. the research methodology. Details explained include the statistical model used, performance measures and various validation techniques. This chapter also dismantles the statistical model used, which the Hidden Markov Models, giving the in-depth understanding of the machine learning algorithms to build up this model. Hence the Viterbi, Forward-Backward algorithms, Markov Chains, Probability calculation and inference means are explained as well.

In **Chapter 5**, the univariate (one attack case) and the multivariate (2 or 3 attack case) results of the work have been explained and shown graphically. The single attack case

shows a trace that is brought if only one attack takes place, so is 2 or 3 attack cases. The results show the effect of these different scenarios. We have concluded that the HMM can give a DDoS silent attack trace and can be incorporated in the cloud computing environment, in turn this can result in VMs to give better performance in the servicing of user request when they are rooted to the VMs.

## **6.2 Discussion of Results**

It is very essential to deal with these DDoS silent attacks and try to detect them as early as possible and get corrected. Thus it is of utmost importance that the various techniques that can be incorporated for this purpose cause. Ultimately this draws us to the exploitation of HMM in handling this DDoS silent attack detection, of which it has shown or proved to be useful.

## **6.3 Application of the Work**

We can conclude that the work in this thesis will be beneficial for the researchers and cloud and/or software professionals:

- The model can be set at proper positions in the cloud for DDoS attack trace detection/s. This will lead to a proper full utilization of virtual machines for their intended use, if silent DDoS attacks are detected and corrected.
- A subset of factors (independent variables) is obtained that can be used to predict any abnormalities that can be done by an attack of certain subset/level/interest.
- Researchers can use machine learning methods rather than traditional detection methods or in combination for a boost, if need be.

## **6.4 Future Work**

This is a study where we find the application of Hidden Markov Model in use, in the detection of DDoS silent attacks in cloud environment, i.e. the effect of HMM in DDoS silent attacks in the cloud. The results provide guidance for future research on the manipulation or adaptation of HMM for other different attack traces in the cloud.

Following are some areas, which we plan to investigate in future:

- In this study, the attacker flooding rate or severity to the targeted virtual machine is not rated or taken into account. The attacks are always not the same and there can be sometimes very serious failures.
- Diverse data sets are welcome for universal results and assessment, this along our pipeline of future planning.

## References

- [1] A. Wess “Computing in the clouds” networker 2007.
- [2] C. Wang, Q. Wang, K. Ren and W. Lou “Ensuring Data Storage Security in Cloud Computing” IEEE-2009.
- [3] X. Tian, X. Wang and A. Zhou “DSP RE-Encryption: A Flexible Mechanism for Access Control Enforcement Management in DaaS” IEEE International Conference on Cloud Computing-2009.
- [4] U. Somani, K. Lakhani and M. Mundra “Implementing Digital Signature with RSA Encryption Algorithm to Enhance the Data Security of Cloud in Cloud Computing” 1<sup>st</sup> International Conference on Parallel, Distributed and Grid Computing (PDGC)-2010.
- [5] Z. Shen and Q. Tong “The Security of Cloud Computing System enabled by Trusted Computing Technology” 2<sup>nd</sup> International Conference on Signal Processing Systems (ICSPPS)-2010.
- [6] A. Sirisha, G. G. Kumari “API Access Control in Cloud Using the Role Based Access Control Model” IEEE-2010.
- [7] M. R. Tribhuvan, V. A. Bhuyar and S. Pirzade “Ensuring Data Storage Security in Cloud Computing through Two-way Handshake based on Token Management” International Conference on Advances in Recent Technologies in Communication and Computing-2010.

- [8] M. Ahmed, Y. Xiang and S. Ali “Above the Trust and Security in Cloud Computing: A Notion towards Innovation” IEEE/IFIP International Conference on Embedded and Ubiquitous Computing-2010.
- [9] Y. Xiao, C. Lin, Y. Jiang, X. Chu and F. Liu “An Efficient Privacy-Preserving Publish-Subscribe Service Scheme for Cloud Computing” IEEE Globecom proceedings-2010.
- [10] M. Menzel, R. Warschofsky, I. Thomas, C. Willems and C. Meinel “The Service Security Lab: A Model-Driven Platform to Compose and Explore Service Security in the Cloud” 6<sup>th</sup> IEEE World Congress on Services-2010.
- [11] A. Albeshri and W. Caelli “Mutual Protection in a Cloud Computing Environment” 12<sup>th</sup> IEEE International Conference on High Performance Computing and Communications-2010.
- [12] G. Zhao, C. Rong, M. G. Jaatuna and F. E. Sandnes “Deployment Models: Towards Eliminating Security Concerns From Cloud Computing” IEEE-2010.
- [13] J. Li, B. Li, L. Meng and D. Sun “HiTrust: A Hybrid Tree based Trust Negotiation Service” 24<sup>th</sup> IEEE International Conference on Advanced Information Networking and Applications Workshops-2010.
- [14] F. Wen and L. xiang “The Study on Data Security in Cloud Computing based on Virtualization” IEEE-2011.
- [15] S. K. Pippal, A. Kumari and D. S. Kushwaha “CTES based Secure approach for Authentication and Authorization of Resource and Service in Clouds” IEEE International Conference on Computer & Communication Technology (ICCCT)-2011.

- [16] Y. Yang and Y. Zhang “A Generic Scheme for Secure Data Sharing in Cloud” International Conference on Parallel Processing Workshops-2011.
- [17] P. Srivastava, S. Singh, A. A. Pinto, S. Verma, V. K. Chaurasiya and R. Gupta “An architecture based on proactive model for security in cloud computing” IEEE International Conference on Recent Trends in Information Technology (ICRTIT)-2011.
- [18] P. Prasad, B. Ojha, R. R. Shahi, R. Lal, A. Vaish and U. Goel “3 Dimensional Security in Cloud Computing” IEEE-2011.
- [19] P. Kalagiakos and M. Bora “Cloud Security Tactics: Virtualization and the VMM” IEEE-2012.
- [20] V. Fusenig and A. Sharma “Security Architecture for Cloud Networking” IEEE International Conference on Computing, Networking and Communications, Cloud Computing and Networking Symposium-2012.
- [21] S. Betgé-Brezetz, G. Kamga, M. Ghorbel and M. Dupont “Privacy Control in the Cloud based on Multilevel Policy Enforcement” 1<sup>st</sup> IEEE International Conference on Cloud Networking (CLOUDNET)-2012.
- [22] P. Jain, D. Rane and S. Patidar “A Novel Cloud Bursting Brokerage and Aggregation (CBBA) Algorithm for Multi Cloud Environment” 2<sup>nd</sup> International Conference on Advanced Computing & Communication Technologies-2012.
- [23] C. Chen, D. J. Guan, Y. Huang and Y. Ou “Attack Sequence Detection in Cloud Using Hidden Markov Model” Seventh Asia Joint Conference on Information Security-2012.



- [24] A. Yu, A. V. Sathanur and V. Jandhyala “A Partial Homomorphic Encryption Scheme for Secure Design Automation on Public Clouds” IEEE-2012.
- [25] K. Mukherjee and G. Sahoo “A Novel Methodology for Security and Privacy of Cloud Computing and its use in e-Governance” IEEE World Congress on Information and Communication Technologies-2012.
- [26] Y. G. Ramaiah and G. Kumari “Efficient Public key Homomorphic Encryption Over Integer Plaintexts” IEEE-2012.
- [27] J. Huang and I. Liao “A Searchable Encryption Scheme for Outsourcing Cloud Storage” IEEE-2012.
- [28] J. Pecarina, S. Pu and J. Liu “SAPPHIRE: Anonymity for Enhanced Control and Private Collaboration in Healthcare Clouds” 4<sup>th</sup> IEEE International Conference on Cloud Computing Technology and Science-2012.
- [29] C. Tang, D. S. Wong, X. Hu and D. Pei “An Efficient Key Distribution Scheme in Cloud Computing” 4<sup>th</sup> IEEE International Conference on Cloud Computing Technology and Science-2012.
- [30] M. Hataba and A. El-Mahdy “Cloud Protection by Obfuscation: Techniques and Metrics” 7<sup>th</sup> International Conference on P2P, Parallel, Grid, Cloud and Internet Computing-2012.
- [31] W. Li, H. Wan, X. Ren and S. Li “A Refined RBAC Model for Cloud Computing” 11<sup>th</sup> IEEE/ACIS International Conference on Computer and Information Science-2012.

- [32] K. Xu, F. Wang and L. Gu “Profiling-as-a-Service in Multi-Tenant Cloud Computing Environments” 32<sup>nd</sup> International Conference on Distributed Computing Systems Workshops-2012.
- [33] T. Chang and H. Meling “Byzantine Fault-Tolerant Publish/Subscribe: A Cloud Computing Infrastructure” 31<sup>st</sup> International Symposium on Reliable Distributed Systems-2012.
- [34] Y. Zhu, H. Hu, G. Ahn, D. Huang and S. Wang “Towards Temporal Access Control in Cloud Computing” 31<sup>st</sup> IEEE Annual International Conference on Computer Communications: Mini-Conference-2012.
- [35] C. Chen, D. J. Guan, Y. Huang and Y. Ou “State-based Attack Detection for Cloud” 2<sup>nd</sup> IEEE International Symposium on Next-Generation Electronics (ISNE) - February 25-26, Kaohsiung, Taiwa-2013.
- [36] I. Lien, Y. Lin, J. Shieh and J. Wu “A Novel Privacy Preserving Location-Based Service Protocol With Secret Circular Shift for K-NN Search” IEEE Transactions on information forensics and security, vol. 8, no. 6, June 2013.
- [37] E. Saleh and C. Meinel “HPISecure: Towards Data Confidentiality in Cloud Applications” 13<sup>th</sup> IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing-2013.
- [38] P. Rewagad and Y. Pawar “Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud
- [39] Yonghee Shin, Andrew Meneely, Laurie Williams, and Jason A. Osborne, “Evaluating Complexity, Code Churn, and Developer Activity Metrics as Indicators of Software Vulnerabilities”

- [40] Ju An Wang, Hao Wang, Minzhe Guo, and Min Xia “Security Metrics for Software Systems”. ACM 2009
- [41] Sree Ram Kumar T, Sumithra A, Alagarsamy K, “The Applicability of Existing Metrics for Software Security”. International Journal of Computer Applications October 2010
- [42] Pratyusa K. Manadhata, Jeannette M. Wing, “An Attack Surface Metric”. IEEE MAY 2011
- [43] Thomas Heumann, Jorg Keller, Sven Turpe. “Quantifying the Attack Surface of a Web Application”, 2010
- [44] Jason L. Wright, Miles McQueen, Lawrence Wellman.” Analyses Of Two End-User Software Vulnerability Exposure Metrics”. IEEE 2012
- [45] National Institute of Standards and Technology, National Vulnerability Database, Common Vulnerability Scoring System Calculator, <http://nvd.nist.gov/cvss.cfm?calculator>.
- [46] The MITRE Corporation, Common Vulnerability and Exposures, CVE List, <http://cve.mitre.org/cve/cve.html>
- [47] Reijo M. Savola, “A Security Metrics Taxonomization Model for Software-Intensive Systems”. Dec 2009
- [48] Reijo M. Savola, “Strategies for Security Measurement Objective Decomposition”. IEEE 2012
- [49] Rosslin John Robles, Young-Sik Jeong, Jong Hyuk Park, Tai-hoon Kim, “Strategy for IT Security in E- Enterprise Environment”. IEEE 2008

- [50] Kitchenham B, Charters S. (2007) Guidelines for performing Systematic Literature Reviews in Software Engineering, Keele University and Durham University Joint report.
- [51] Jesus Luna, Hamza Ghani, Daniel Germanus and Neeraj Suri “A security metrics framework for the cloud”
- [52] Jennifer Bayuk “Cloud Security Metrics” IEEE, 2011
- [53] National Institute of Standards and Technology (NIST), 2013. “NIST Cloud Computing Reference Architecture Cloud Service Metrics Description,” NIST Draft Publication, April 3 – [http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/RATax\\_CloudMetrics\\_Meeting\\_04112013](http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/RATax_CloudMetrics_Meeting_04112013).
- [54] K. Wan, X. Gao, X. Liu and S. Cui “A Cloud Cooperative Attack System for Networking Anti-stealth Combat”-IEEE -2013