

**A CLASSIFIER MECHANISM FOR HOST-  
BASED INTRUSION DETECTION AND  
PREVENTION SYSTEM IN CLOUD  
COMPUTING ENVIRONMENT**

**AWS NASER JABER AL-ZARQAWEE**

**UMP**

**DOCTOR OF PHILOSOPHY**

**UNIVERSITI MALAYSIA PAHANG**

## UNIVERSITI MALAYSIA PAHANG

### DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : Aws Naser Jaber AL-Zarqawee

Date of Birth : 08/04/1981

Title : **A CLASSIFIER MECHANISM FOR HOST BASED  
INTRUSION DETECTION AND PREVENTION  
SYSTEM IN CLOUD COMPUTING ENVIRONMENT**

Academic Session : 2017/2018

I declare that this thesis is classified as:

- CONFIDENTIAL** (Contains confidential information under the Official Secret Act 1997)\*
- RESTRICTED** (Contains restricted information as specified by the organization where research was done)\*
- OPEN ACCESS** I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

\_\_\_\_\_  
(Student's Signature)

\_\_\_\_\_  
(Supervisor's Signature)

A10355939  
New IC/Passport Number  
Date:

T.S Dr. Mohamad Fadli Zolkipli  
Name of Supervisor  
Date:

NOTE : \* If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

## **SUPERVISOR'S DECLARATION**

We hereby declare that we have checked this thesis, and, in our opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Doctor of Philosophy

---

(Supervisor's Signature)

Full Name : T.S DR. MOHAMAD FADLI ZOLKIPLI

Position : SENIOR LECTURER

Date : / /2018

---

(Co-supervisor's Signature)

Full Name : DR. MAZLINA BINTI ABDUL MAJID

Position : ASSOC. PROFESSOR

Date : / /2018

### STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

---

(Student's Signature)

Full Name : AWS NASER JABER AL-ZARQAWEE

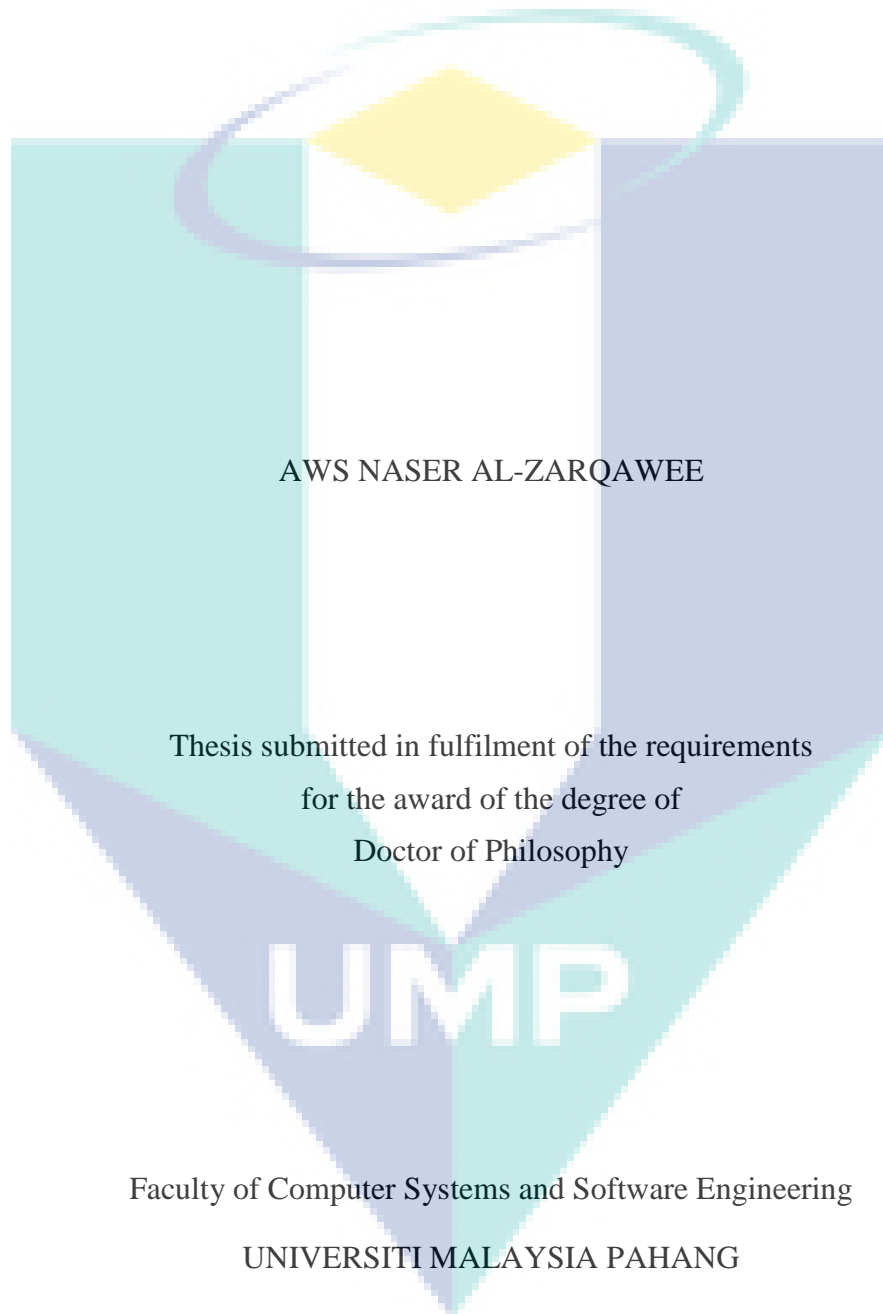
ID Number : PCC13011

Date : / /2018



UMP

A CLASSIFIER MECHANISM FOR HOST BASED INTRUSION DETECTION  
AND PREVENTION SYSTEM IN CLOUD COMPUTING ENVIRONMENT



AWS NASER AL-ZARQAWEE

Thesis submitted in fulfilment of the requirements  
for the award of the degree of  
Doctor of Philosophy

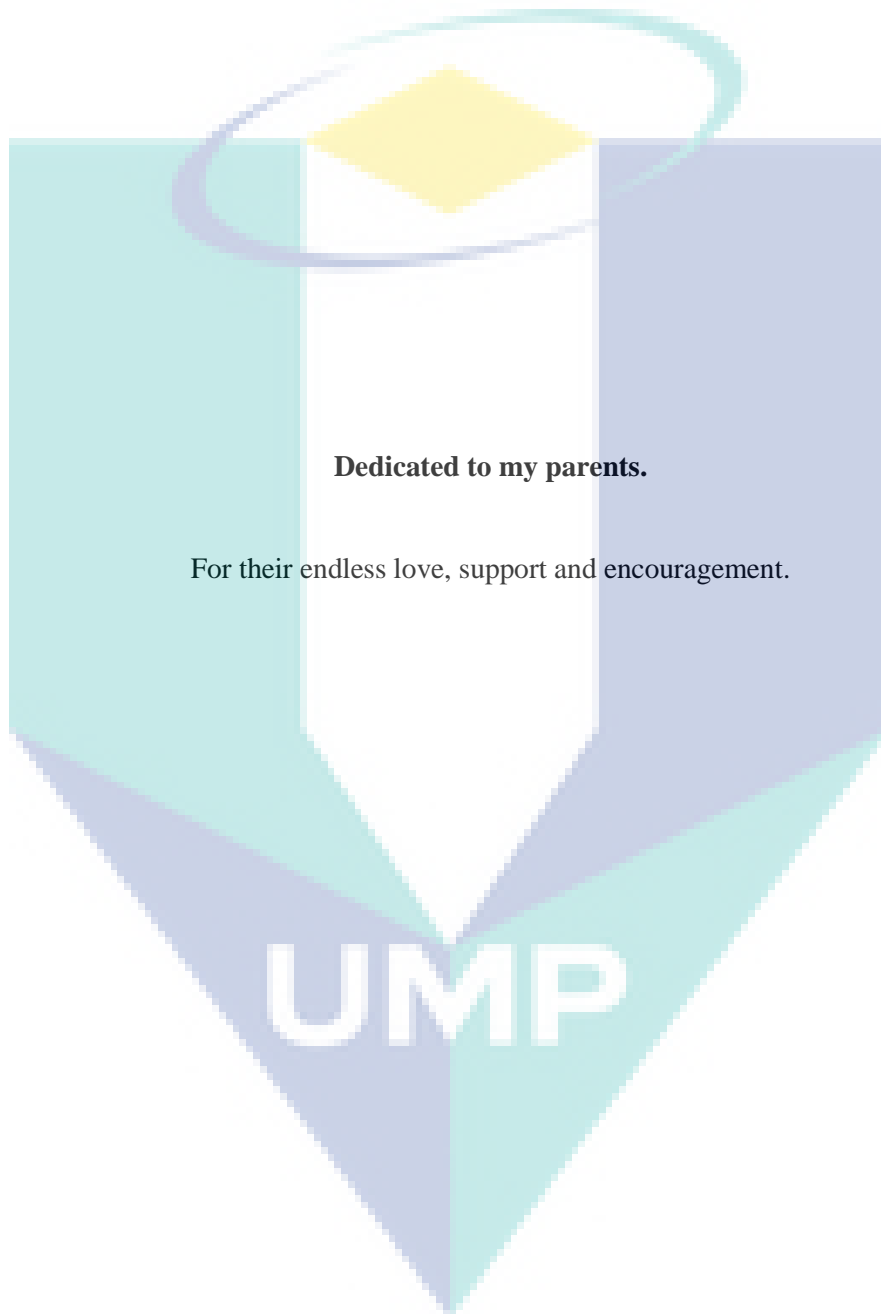
UMP

Faculty of Computer Systems and Software Engineering

UNIVERSITI MALAYSIA PAHANG

SEPTEMBER 2018

## DEDICATION



## ACKNOWLEDGEMENTS

Alhamdulillah with the will of Allah, I have successfully completed this research. Without the strength applied to me, I would not be able to finish this subject field on time devoted. This thesis is prepared to fulfil the requirements for Doctor of Philosophy from Faculty of Computer Systems & Software Engineering, Universiti Malaysia Pahang

I would like to take this opportunity to convey my sincere thanks and deepest gratitude to Dr. Mohamad Fadli Zolkipli and Associate Professor Dr. Mazlina Abdul Majid for all their help and valuable guidance provided to me during the preparation of this thesis.

I consider myself privileged to have had the opportunity to work under his guidance. Moreover, I would like to dedicate this thesis to the dearest ones, my wife Zena for her patience and the encouragement he provided me with during the entire period of the study, and my mother who shared the stress in my life, encouraged me in times of dismay, cheered me up in times of distress, and renewed my hope in times of despair



UMP

## ABSTRAK

Serangan Distributed Denial-of-Service (DDoS) adalah insiden yang sering berlaku dalam persekitaran pengkomputeran awam yang menyebabkan gangguan utama prestasi. Sistem pengesanan dan pencegahan pencerobohan (IDP) adalah merupakan alat untuk melindungi daripada sebarang insiden tersebut, dan penempatan sistem ID/IP yang tepat pada rangkaian adalah sangat penting untuk pemantauan yang optimum dan mencapai keberkesanan yang maksimum dalam melindungi sistem. Walaupun dengan adanya sistem tersebut, tahap keselamatan pengkomputeran awam mesti dipertingkatkan. Semakin banyak serangan yang lebih kuat cuba untuk mengawal persekitaran pengkomputeran awam tersebut; serangan tersebut adalah termasuk hyperjacking mesin-maya (VM) dan juga ancaman keselamatan rangkaian tradisional seperti pengintipan trafik (memintas trafik rangkaian), pengintipan alamat dan pemalsuan VM atau alamat IP. Menguruskan IDPS berasaskan hos (H-IDPS) adalah sangat sukar kerana maklumat perlu dikongfigurasi dan diuruskan oleh setiap hos, ianya penting untuk memastikan penganalisis keselamatan dapat memahami struktur rangkaian sepenuhnya bagi membezakan antara positif palsu dan masalah sebenar. Untuk tujuan tersebut, adalah sangat penting untuk memahami pengelas paling utama dalam pembelajaran mesin, kerana ianya menawarkan perlindungan terhadap penggera positif palsu dalam serangan DDoS. Bagi merancang lebih banyak pengelasan yang berkesan, sistem bagi menilai pengelas perlu dibangunkan. Dalam thesis ini, mekanisme reka bentuk untuk pengelas H-IDPS dalam persekitaran pengkomputeran awam telah dibangunkan. Reka bentuk mekanisme ini berdasarkan Optimasi Antlion hibrid Algoritma (ALO) dengan Multilayer Perceptron (MLP) untuk berlindung dari serangan DDoS. Untuk melaksanakan mekanisme yang dicadangkan, kami menunjukkan kekuatan pengelas menggunakan satu dataset yang dikurangkan dimensi menggunakan NSL-KDD. Selain itu, kami memberi tumpuan terperinci kepada kajian dataset NSL-KDD yang mengandungi hanya rekod terpilih. Dataset yang dipilih ini menyediakan analisis yang baik terhadap pelbagai teknik pembelajaran mesin untuk H-IDPS. Penilaian terhadap Sistem H-IDPS ini menunjukkan peningkatan ketepatan pengesanan pencerobohan dan mengurangkan penggera positif palsu berbanding hasil kajian lain yang berkaitan. Ini dapat digambarkan dengan menggunakan teknik matriks kekeliruan untuk mengatur pengelas, menggambarkan prestasi dan menilai tingkah laku secara keseluruhan.



## ABSTRACT

Distributed denial-of-service (DDoS) attacks are incidents in a cloud computing environment that cause major performance disturbances. Intrusion-detection and prevention system (IDPS) are tools to protect against such incidents, and the correct placement of ID/IP systems on networks is of great importance for optimal monitoring and for achieving maximum effectiveness in protecting a system. Even with such systems in place, however, the security level of general cloud computing must be enhanced. More potent attacks attempt to take control of the cloud environment itself; such attacks include malicious virtual-machine (VM) hyperjacking as well as traditional network-security threats such as traffic snooping (which intercepts network traffic), address spoofing and the forging of VMs or IP addresses. It is difficult to manage a host-based IDPS (H-IDPS) because information must be configured and managed for every host, so it is vital to ensure that security analysts fully understand the network and its context in order to distinguish between false positives and real problems. For this, it is necessary to know the current most important classifiers in machine learning, as these offer feasible protection against false-positive alarms in DDoS attacks. In order to design a more efficient classifier, it is necessary to develop a system for evaluating the classifier. In this thesis, a new mechanism for an H-IDPS classifier in a cloud environment has desigend. The mechanism's design is based on the hybrid Antlion Optimization Algorithm (ALO) with Multilayer Perceptron (MLP) to protect against DDoS attacks. To implement the proposed mechanism, we demonstrate the strength of the classifier using a dimensionally reduced dataset using NSL-KDD. Furthermore, we focus on a detailed study of the NSL-KDD dataset that contains only selected records. This selected dataset provides a good analysis of various machine-learning techniques for H-IDPS. The evaluation process H-IDPS system shows the increases of intrusion detection accuracy and decreases the false positive alarms when compared to other related works. This is epitomized by the skilful use of the confusion matrix technique for organizing classifiers, visualizing their performance, and assessing their overall behaviour.

The logo for UMP (Universiti Malaysia Perlis) is a large, stylized letter 'V' shape. The left side of the 'V' is light blue, and the right side is light green. The letters 'UMP' are written in white, bold, sans-serif font across the center of the 'V'.

# TABLE OF CONTENT

<b>DECLARATION</b>	
<b>TITLE PAGE</b>	
<b>ACKNOWLEDGEMENTS</b>	<b>iii</b>
<b>ABSTRAK</b>	<b>iv</b>
<b>ABSTRACT</b>	<b>v</b>
<b>TABLE OF CONTENT</b>	<b>vi</b>
<b>LIST OF FIGURES</b>	<b>xi</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xiii</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Motivation	1
1.2 Problem Statement	3
1.3 Research Aim and Objectives	6
1.4 Research Scope	7
1.5 Research Framework	7
1.6 Thesis Outline	10
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>11</b>
2.1 Overview	11
2.2 Cloud Computing Security and DDoS Attack Classifiers	11
2.2.1 Cloud Computing	13
2.2.2 Data Storage Security in Cloud Computing	15
2.2.2.1 Cloud Storage	15

2.2.3	Cloud DDoS Machine Learning Techniques for Classification of Attacks	19
2.2.3.1	ANNs	25
2.2.3.2	MLP	27
2.2.3.3	K-Nearest Neighbours	28
2.2.3.4	Fuzzy Logic	29
2.2.3.5	Evolutionary Computation	29
2.2.3.6	Probabilistic Reasoning	30
2.3	DDoS Benchmark Dataset for Verification of Machine Learning Classifiers	35
2.3.1	NSL-KDD	36
2.3.2	DARPA Family	36
2.3.3	CAIDA	36
2.3.3.1	Dataset Dimension Reduction	37
2.4	Evaluating Findings for Existing H-IDPS Cloud DDoS Attack Classifiers	41
2.5	Chapter Summary	41
<b>CHAPTER 3 METHODOLOGY</b>		<b>42</b>
3.1	Overview	42
3.2	Methodology Design Process	42
3.2.1	Design of Classifier Mechanism	44
3.2.1.1	ALO Process	44
3.2.1.2	ALO-MLP Classifier Mechanism	48
3.2.2	Implementing ALO-MLP as a Classifier for the NSL-KDD Dataset	52
3.2.2.1	Scenario 1: Denial of Service	55
3.2.2.2	Scenario 2: Probing	55
3.2.2.3	Scenario 3: R2L	55
3.2.2.4	Scenario 4: User to Root	55
3.2.3	Performance evaluation of the Proposed Mechanism	55
3.2.3.1	Accuracy	58
3.2.3.2	Incorrect Classification Rate	58

3.2.3.3	Confusion Matrix	58
3.2.3.4	Precision	59
3.2.3.5	FN Rate	59
3.2.3.6	Recall	60
3.2.3.7	F1 Score	60
3.2.3.8	TPR	60
3.2.3.9	Area Under ROC Curve (AUC)	61
3.2.3.10	Matthews Correlation Coefficient	62
3.3	Chapter Summary	62
<b>CHAPTER 4 IMPLEMENTATION AND RESULTS</b>		<b>63</b>
4.1	Overview	63
4.2	Implementation Phases	63
4.2.1	Implementation of Designed ALO-MLP Classifier	64
4.2.1.1	Flood Pre-Processor Data Structure	65
4.2.1.2	Cloud Environment H-IDPS Network System Specifications	67
4.2.2	Implementation of ALO-MLP H-IDPS Classifier using NSL-KDD	68
4.3	Evaluation of Mechanism	70
4.3.1	Parameters	70
4.3.2	Confusion Matrix	71
4.4	ALO-MLP classifier Scenario Results Through NSL-KDD	71
4.4.1	DoS Scenario Results	71
4.4.2	U2R Scenario Results	72
4.4.3	R2L Scenario Results	73
4.4.4	Probes Scenario Results	74
4.5	Variance Blacklist H-IDPS	75
4.6	Chapter Summary	76

<b>CHAPTER 5 EVALUATION AND COMPARATIVE ANALYSIS</b>	<b>77</b>
5.1 Overview	77
5.2 Evaluation of H-IDPS Snort	77
5.2.1 Snort with ALO-MLP + PHAD	78
5.2.2 Snort with ALO-MLP + PHAD + ALAD	79
5.2.3 Snort with ALO-MLP+ ALAD + LERAD	80
5.3 Evaluation of ALO-MLP in Comparison with Most Common Classifiers	81
5.4 Comparative Analysis for ALO-MLP Classifier with Other Classifier Mechanisms	83
5.4.1 DoS comparison	83
5.4.2 Probe comparison	86
5.4.3 R2L comparison	89
5.4.4 U2R comparison	92
5.5 Chapter Summary	95
<b>CHAPTER 6 CONCLUSION</b>	<b>96</b>
6.1 Overview	96
6.2 Contribution	96
6.3 Future Works	97
<b>REFERENCES</b>	<b>98</b>
<b>APPENDIX A</b>	<b>106</b>
<b>APPENDIX B</b>	<b>107</b>
<b>APPENDIX C</b>	<b>108</b>

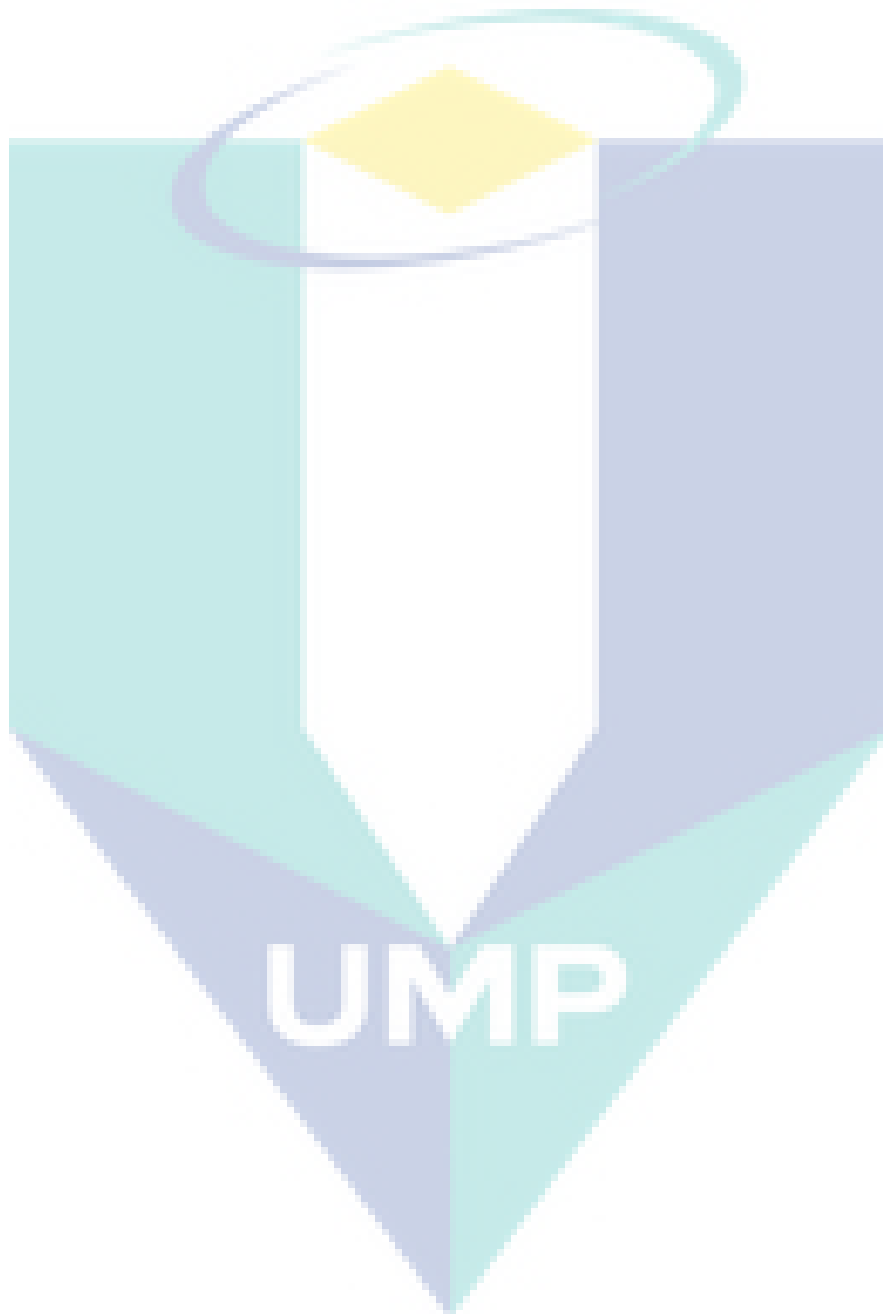
## LIST OF TABLES

Table 2.1	Cloud storage advantages and disadvantages	17
Table 2.2	DDoS attack types	22
Table 2.3	Comparison of H-IDPS methods based on collected criteria	23
Table 2.4	H-IDPS machine learning methods.	25
Table 2.5	Summary of classification techniques.	31
Table 2.6	Popular NSL-KDD classification approaches based on feature selection and classifier method.	35
Table 2.7	Summary of PCA approaches.	39
Table 2.8	Summary of LDA approaches.	40
Table 2.9	Comparison of studies that classified the NSL-KDD dataset in terms of overall accuracy	41
Table 4.1	System specifications	67
Table 4.2	NSL-KDD features.	69
Table 4.3	Attack type and their related attack.	69
Table 4.4	Control parameters used in H-IDPS.	70
Table 4.5	Sample confusion matrix for ALO-MLP for 74,637 samples.	71
Table 4.6	ALO-MLP classifier for DoS over metrics.	72
Table 4.7	ALO-MLP classifier for U2R over metrics	73
Table 4.8	ALO-MLP Classifier for R2L over metrics.	74
Table 4.9	ALO-MLP Classifier for Probe over metrics	75
Table 5.1	DoS comparison with other related works for accuracy, incorrect classification rate, FN , TPR , precision, recall and F1 score.	85
Table 5.2	Probe comparison with other related works for accuracy, incorrect classification rate, FN, TPR, precision, recall and F1 score.	88
Table 5.3	R2L comparison with other related works for accuracy, incorrect classification rate, FN, TPR, precision, recall and F1 score	91
Table 5.4	U2R comparison with other related works for accuracy, incorrect classification rate, FN, TPR, precision, recall and F1 score	94

## LIST OF FIGURES

Figure 1.1	DDoS attack vector frequency Q1.	3
Figure 1.2	Drawbacks of cloud computing security.	5
Figure 1.3	Operational research Framework.	9
Figure 2.1	Cloud computing.	13
Figure 2.2	Type-1 native bare metal hypervisor.	14
Figure 2.3	Type-2 hosted hypervisor.	14
Figure 2.4	DDoS attack scenario in the cloud computing environment.	20
Figure 2.5	Black Lotus: three main DDoS flooding attacks.	21
Figure 2.6	Schematic of H-IDPS.	22
Figure 2.7	Classification of true/false negative/ positive.	24
Figure 2.8	Structure of ANN.	26
Figure 2.9	Schematic of MLP used in IDPS.	27
Figure 3.1	Proposed methodology and relation to research objectives.	43
Figure 3.2	Classifier mechanism design.	44
Figure 3.3	Operators of the ALO algorithm.	45
Figure 3.4	ALO process flowchart.	47
Figure 3.5	ALO-MLP mechanism.	49
Figure 3.6	Detection and prevention engine in Snort.	50
Figure 3.7	ALO-MLP as a pre-processor classifier in Snort core.	51
Figure 3.8	Implementing ALO-MLP.	52
Figure 3.9	ALO-MLP classifier testing in Weka.	54
Figure 3.10	Evaluation Metrics	57
Figure 3.11	Confusion matrix classification.	59
Figure 3.12	Illustration of the area under the AUC curve.	61
Figure 4.1	DDoS pre-processor key data structure.	65
Figure 4.2	DDoS pre-processor key data structure.	66
Figure 4.3	Two HP servers.	67
Figure 4.4	UDP highest and lowest IP attack rates.	75
Figure 4.5	TCP highest and lowest IP attack rates.	76
Figure 5.1	Daily DDoS prevention levels for Snort H-IDPS.	78
Figure 5.2	Daily DDoS prevention levels for Snort H-IDPS using ALO-MLP with and without PHAD.	79
Figure 5.3	Daily DDoS prevention levels for Snort H-IDPS using ALO-MLP compared with PHAD and ALAD.	80


Figure 5.4	Daily DDoS prevention levels for Snort H-IDPS using ALO-MLP classifier compared with ALAD and LERAD.	81
Figure 5.5	ALO-MLP Compared with Most Common Classifiers	82





## LIST OF ABBREVIATIONS

ALAD	Application Layer Anomaly Detection
ALO	Antlion Optimization Algorithm
ANN	Artificial Neural Network
BPNN	Backpropagation Neural Network
CPU	Central Processing Unit
DARPPA	Defence Advanced Research Projects Agency
DDoS	Denial-Of-Service Attack
DNS	Domain Name System
EV	Evolutionary Computation
FDR	Fisher's Discriminant Ratio
FL	Fuzzy Logic
FN	False Negative
FP	False Positive
GA	Genetic Algorithm
GAR-forest	Greedy randomized adaptive search procedure with annealed randomness
GRASP	Greedy Randomized Adaptive Search Procedure
H-IDPS	Host-Based Intrusion Detection and Prevention System
HTTP	Hypertext Transfer Protocol
IASS	Infrastructure as A Service
ICMP	Internet Control Message Protocol
IDPS	Intrusion Detection System and Prevention System
IDS	Intrusion Detection System
IP	Internet Protocol Address
IPS	Intrusion Prevention Systems
IT	Information Technology
KNN	K-Nearest Neighbours
KVM	Kernel-Based Virtual Machine
LAN	Local Area Network
LDA	Linear Discriminant Analysis
LERAD	Learning Rules for Anomaly Detection



MLF	Multi-Level Fuzzy Min-Max Neural Network
MLP	Multilayer Perceptron
NN	Neural Network
NTP	Network Time Protocol
OTN	Option List
PASS	Platform as A Service
PCA	Principal Component Analysis
PCAP	Packet Capture
PHAD	Packet Header Anomaly Detection
QoS	Quality of Service
R2L	Remote to Local Attack
RQs	Research Questions
SAAS	Software as A Service
SU	Symmetrical Uncertainty
SVM	Support Vector Machine
SYN	Synchronize
TN	True Negatives
TP	True Positive
TTL	Time to Live
U2R	User to Root Attack
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
VLAN	Virtual LAN
VMs	Virtual Machines
XML	Extensible Markup Language

## CHAPTER 1

### INTRODUCTION

#### 1.1 Motivation

In the current cyber environment, the importance of cyber security cannot be denied. As the size and reach of the Internet continues to grow, cyber security has become a necessity for large, well-known organizations, small businesses, and individuals. Intrusion Detection and Prevention Systems (IDPS) are an efficient way of detecting and preventing cyber security threats (Modi & Acha, 2017). However, not enough attention and awareness has been placed on IDPS, especially among small businesses and individuals. As a result, the selection and deployment of IDPS is widely regarded as being highly technical, expensive, and time-consuming.

Today, the research community is focused on finding appropriate solutions for the issue of cloud security, which is the biggest obstacle preventing the full adoption of cloud services. In the world of computing, there are different security domains, each of which addresses various aspects of security (Kritikos et al., 2017). However, mixed in with these domains are numerous security challenges that need to be addressed and handled. A survey conducted by Right Scale in January 2017 asked 1002 IT professionals about the adoption of cloud infrastructure and its related technology. Security and lack of resources were identified as the major problems, with 25% of the IT professionals considering security to be a major obstacle to the adoption of cloud computing (Birje et al., 2017).

All types of attacks that are applicable to computer networks and data in transit similarly apply to the cloud computing paradigm, such as Remote to Local (R2L), Probe,

User to Root (U2R), Side channel, Masquerade, DNS spoofing, SQL injections, and Distributed Denial-of-Service (DDoS) attacks (Latha & Prakash, 2017).

To handle security issues in the cloud environment, techniques based on security policies and firewalls have been proposed. While these are primary security techniques, they are not sufficient to provide secure systems. For instance, a firewall sniffs network packets at the boundary of the network to detect and prevent attacks from entering the network, but it cannot detect insider attacks and does not provide in-depth packet analysis. Moreover, attacks such as DDoS are too complex to be discovered using traditional firewalls or countermeasures.

Lin and Li (2018) investigated the top four infrastructure DDoS attacks. As shown in Figure 1.1, User Datagram Protocol (UDP) fragments, Domain Name System (DNS) floods, Network Time Protocol (NTP) floods, and Charge attacks are the dominant threats. Compared to the previous quarter, there was a rise in UDP fragment, NTP, and Charge attacks, and a slight fall in DNS attacks. Organizations can keep their servers safe from these DDoS attacks if services such as Charge and NTP are inaccessible from the Internet or are patched.



UMP

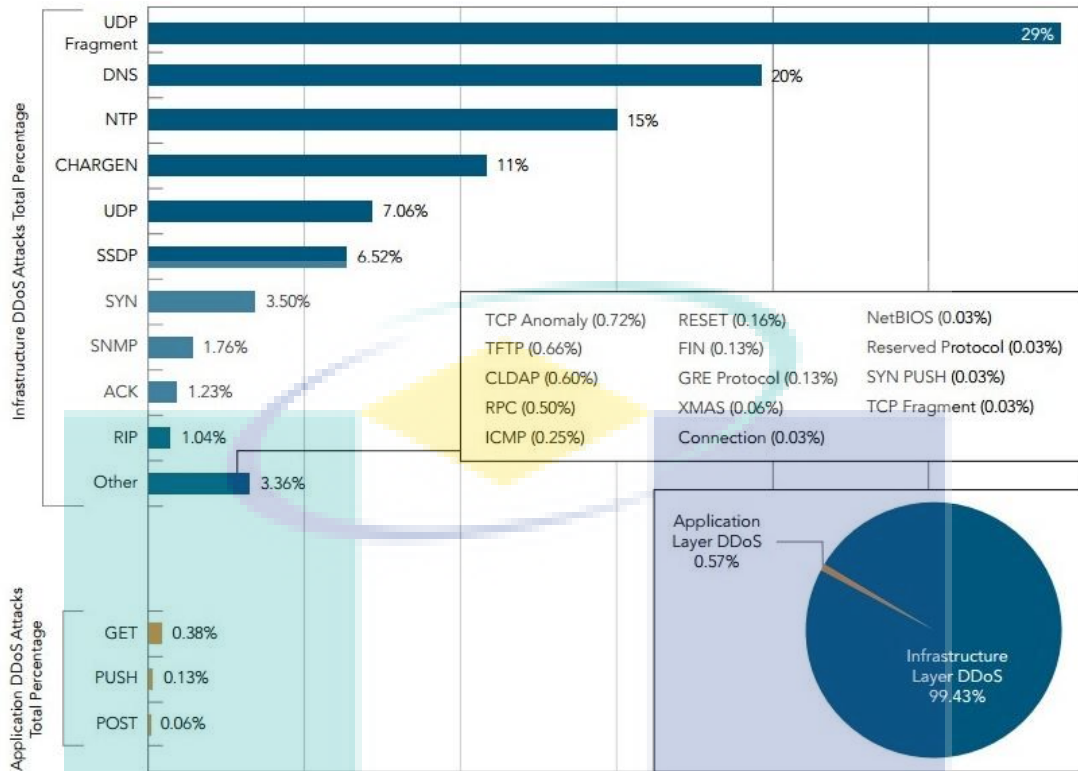


Figure 1.1 DDoS attack vector frequency Q1.

Source: Lin and Li (2018)

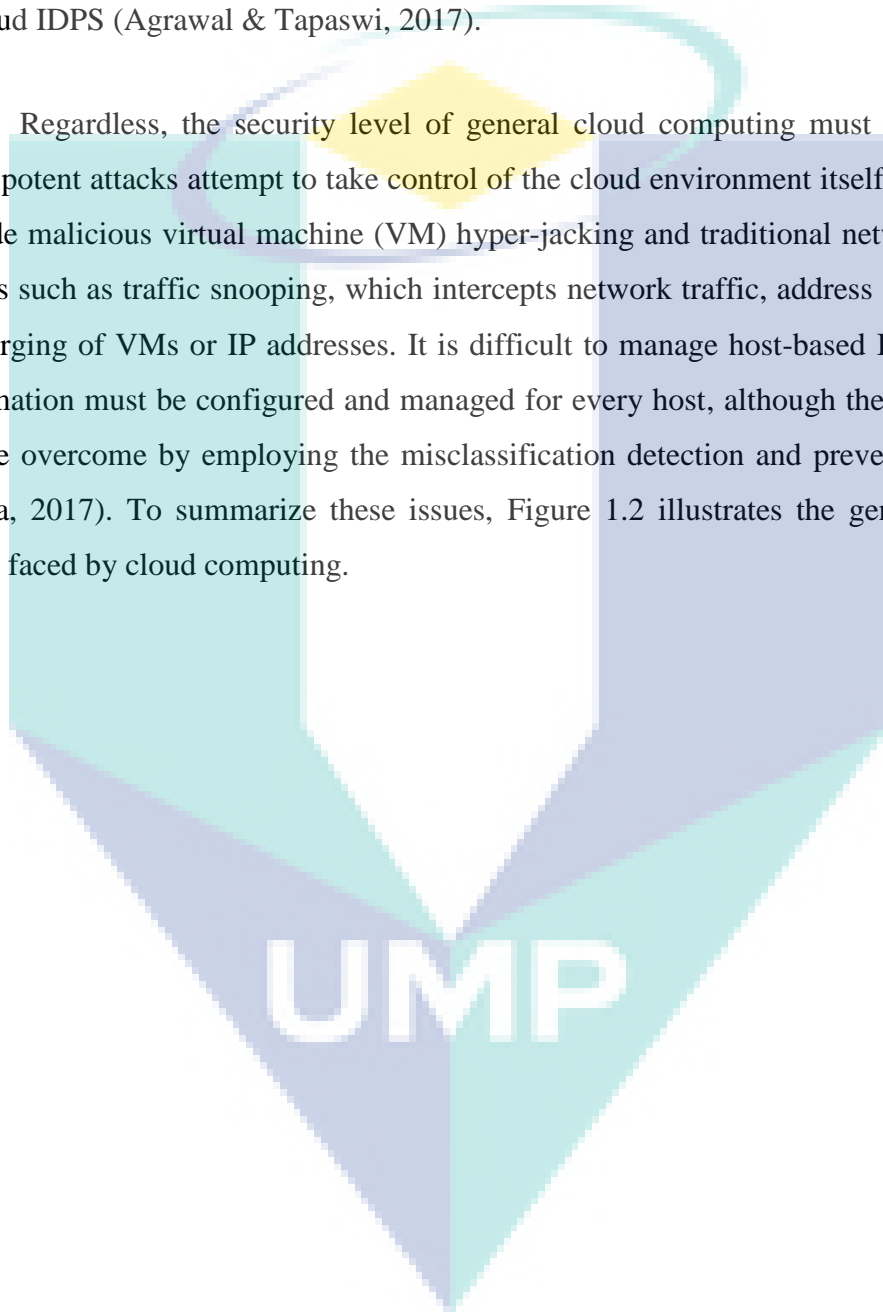
In this research, a classifier mechanism is developed for host-based IDPS in the cloud environment. The mechanism is designed based on the Antlion Optimization Algorithm (ALO) with a Multilayer Perceptron (MLP) to protect against DDoS attacks. To verify the proposed mechanism, we demonstrate the strength of the classifier using a dimensionally reduced dataset (Su et al., 2017). The mechanism is then evaluated under NSL-KDD DDoS traffic conditions. The proposed mechanism is compared with mechanisms developed in the past five years that have been reported in the literature.

## 1.2 Problem Statement

The volume of targeted network attacks is steadily increasing and evolving, forcing businesses to revamp their network security systems due to possible data and financial losses. Intrusion Detection and Prevention Systems (IDPS) is an essential component for any security system. IDPS main function is to identify unauthorized access that attempts to compromise confidentiality, integrity or availability of computer or computer networks. One of the major steps in encountering the problem of IDPS is classifying the types of attacks (Modi & Acha, 2017).

The rapid growth and increasing utility of the Internet mean that Internet security issues are of vital importance. DDoS attacks are one of the most serious issues, and a means of preventing such attacks should be devised as soon as possible. These attacks prevent users from communicating with service providers and have damaged many major websites around the world. Flooding attacks present a significant problem to cloud IDPS (Agrawal & Tapaswi, 2017).

Regardless, the security level of general cloud computing must be enhanced. More potent attacks attempt to take control of the cloud environment itself; such attacks include malicious virtual machine (VM) hyper-jacking and traditional network security threats such as traffic snooping, which intercepts network traffic, address spoofing, and the forging of VMs or IP addresses. It is difficult to manage host-based IDPS because information must be configured and managed for every host, although these drawbacks can be overcome by employing the misclassification detection and prevention method (Kizza, 2017). To summarize these issues, Figure 1.2 illustrates the general security issues faced by cloud computing.



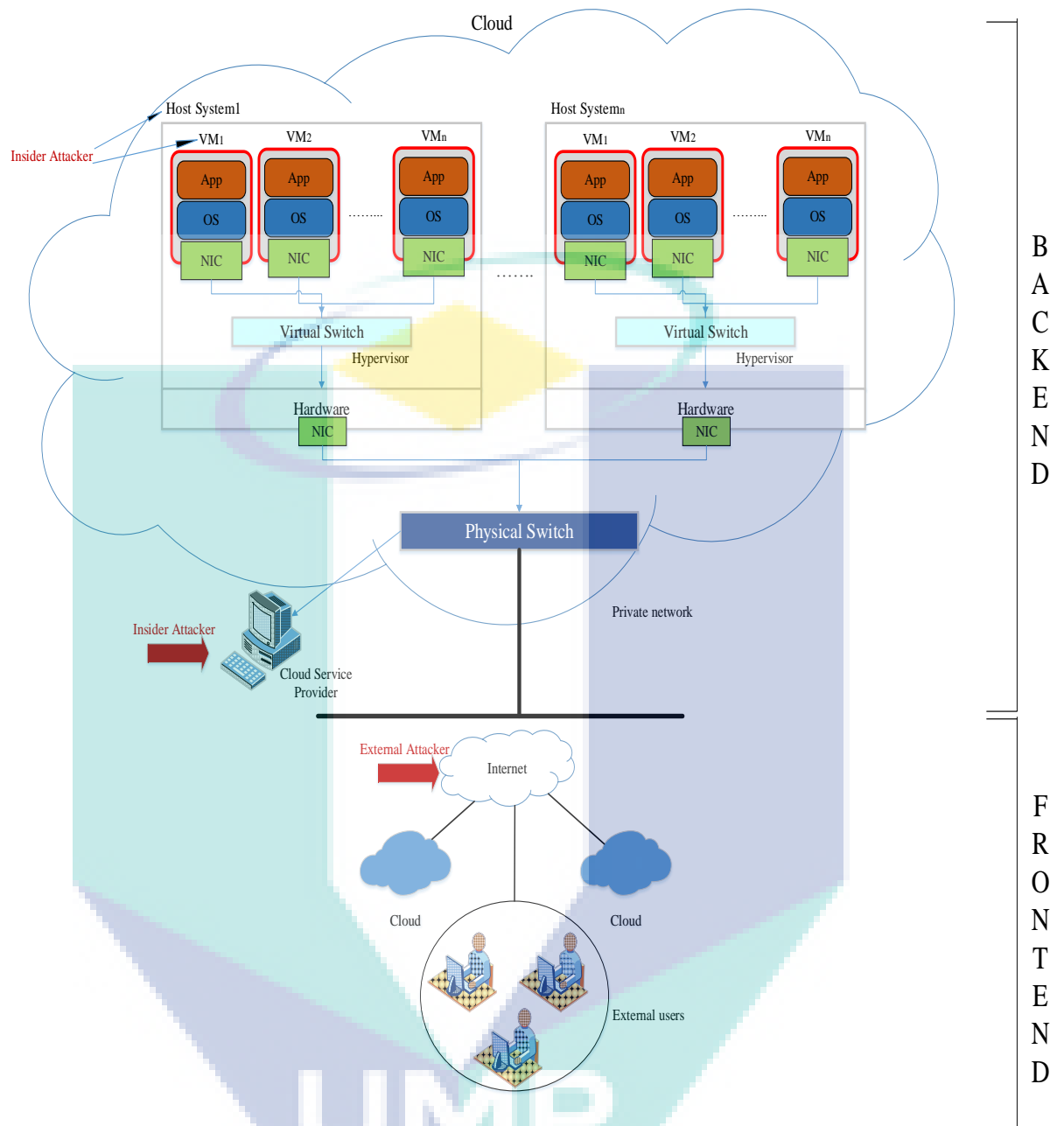


Figure 1.2 Drawbacks of cloud computing security.

To date, various methods have been proposed to prevent a user’s connection being cut-off when trying to perform an action that is flagged as a malicious activity in the IDPS. IT departments must then spend a significant amount of time checking every computer that encounters a false negative scenario (Patel et al., 2013).

Even though, the pre-processing classification in H-DIPS is made successfully, classification of DDoS Datasets seems to be a challenging task in term of lower accuracy and false alarm rates (Eid et al., 2011; Hassanien et al., 2014; Emiro De la Hoz et al., 2014; Enache & Patriciu, 2014; Kanakarajan & Muniasamy, 2016; Pajouh et al., 2017).

Hence, these issues led to the fundamental research questions examined in this thesis are:

- RQ1: Do current classifier mechanisms offer feasible accuracy, and false negative rate against DDoS attacks?
- RQ2: How efficient is the development of current classifiers against DDoS traffic in terms of the host-based intrusion detection and prevention systems?
- RQ3: What is the overall performance of the evaluation procedures used to assess the results in terms of their metrics and comparative analysis?

### **1.3 Research Aim and Objectives**

The purpose of this research is to design a new classifier for host-based intrusion detection and prevention system in the cloud environment to achieve the better performance in term of accuracy, incorrect classification rate, false negate rate, true positive rate, precision, recall, F1 score and area under curve during the new classifier placement. This aim be further explicated by the following specific research objectives:

- i. To design a classifier mechanism using ALO-MLP to improve the accuracy in host-based intrusion detection and prevention system and reduce the false negative rate.
- ii. To develop the ALO-MLP classifier mechanism through DDoS traffic as host based in intrusion detection and prevention system using NLS-KDD dataset.
- iii. To evaluate the performance of the proposed mechanism using machine learning metrics and compare it with existing classifiers in simulation environment.



## 1.4 Research Scope

The main domain of this research is implement a classifier mechanism for Intrusion detection and prevention systems in cloud computing Environment. The proposed mechanism uses a new designed ALO with MLP for increase the accuracy rate and decrease the false alarm rate. Snort has used as H-IDPS and the network traffic came from extracting features of DDoS dataset NSL-KDD. Furthermore, a new classifier has tested over cloud environment using hypervisor, and that done through configure Snort on it. Moreover, this proposed mechanism will improve the problem of low accuracy and high false alarm to see the best performance of H-IDPS. The designed mechanism to be developed both in semi real time and simulated environment.

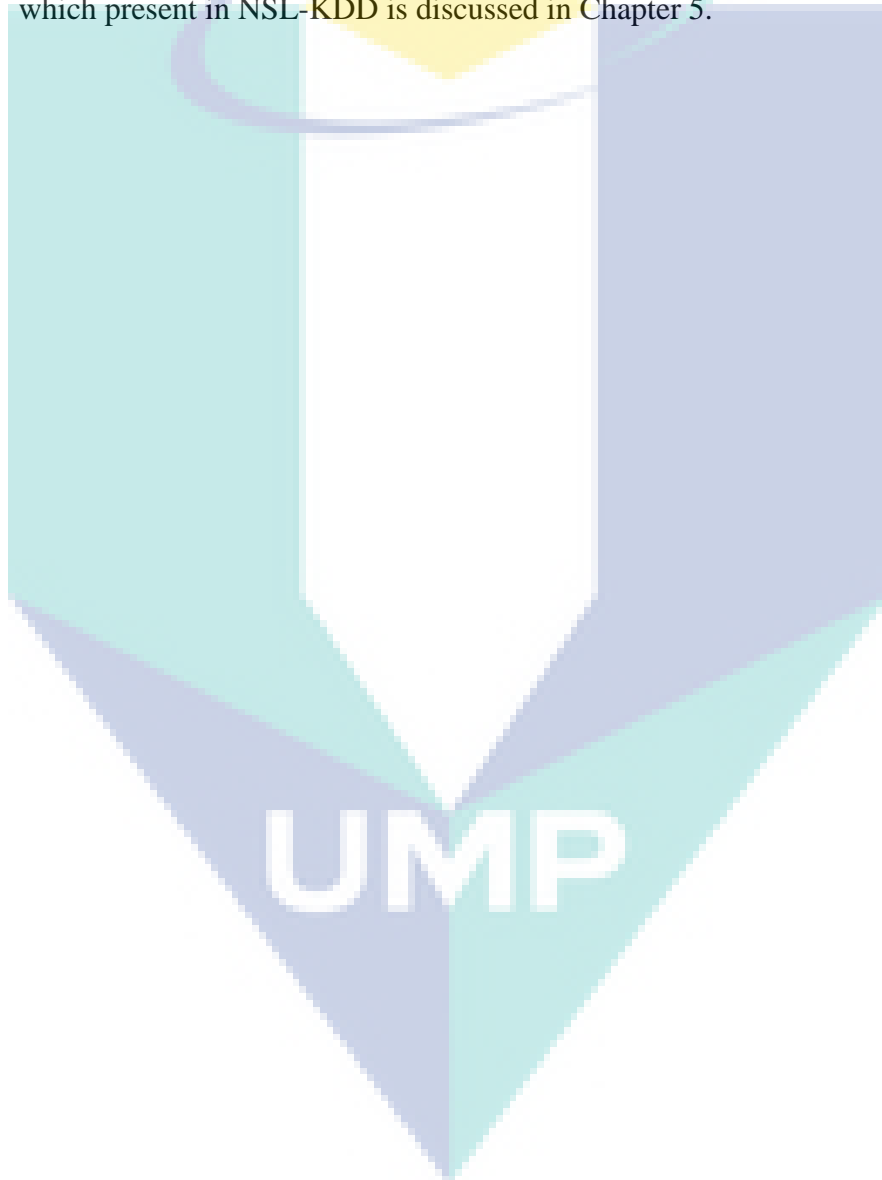
## 1.5 Research Framework

The overall process mechanism used throughout this research is illustrated in Figure 1.3. The operational framework is divided across the following stages: literature review, research methodology, and the evaluation stage for the proposed strategy.

These stages of the research operational framework can be described as follows:

1. A comprehensive investigation of the security threats in cloud computing through a review of the available relevant literature. Furthermore, a comprehensive literature review of existing H-IDPS examines the applicability of deploying detection–prevention methods in cloud computing. The literature review conducted in this research is summarized in Chapter 2.
2. Design and implement a new classifier based-on ALO-MLP in Snort and investigation of the core mechanism for the deployment of H-IDPS in the cloud environment. The aim of these mechanism is to enable the prevention of DDoS Attacks through NSL-KDD DDoS dataset, and this is discussed in Chapter 3.

3. The resulting datasets are evaluated through several machine learning metrics using Weka as a simulation platform, as demonstrated in Chapter 4.
4. Mechanism evaluation using our prevention mechanism metrics with other machine learning metrics, for example, correct and incorrect classification. Further, a comparative study with other mechanisms based on DDoS attacks which present in NSL-KDD is discussed in Chapter 5.



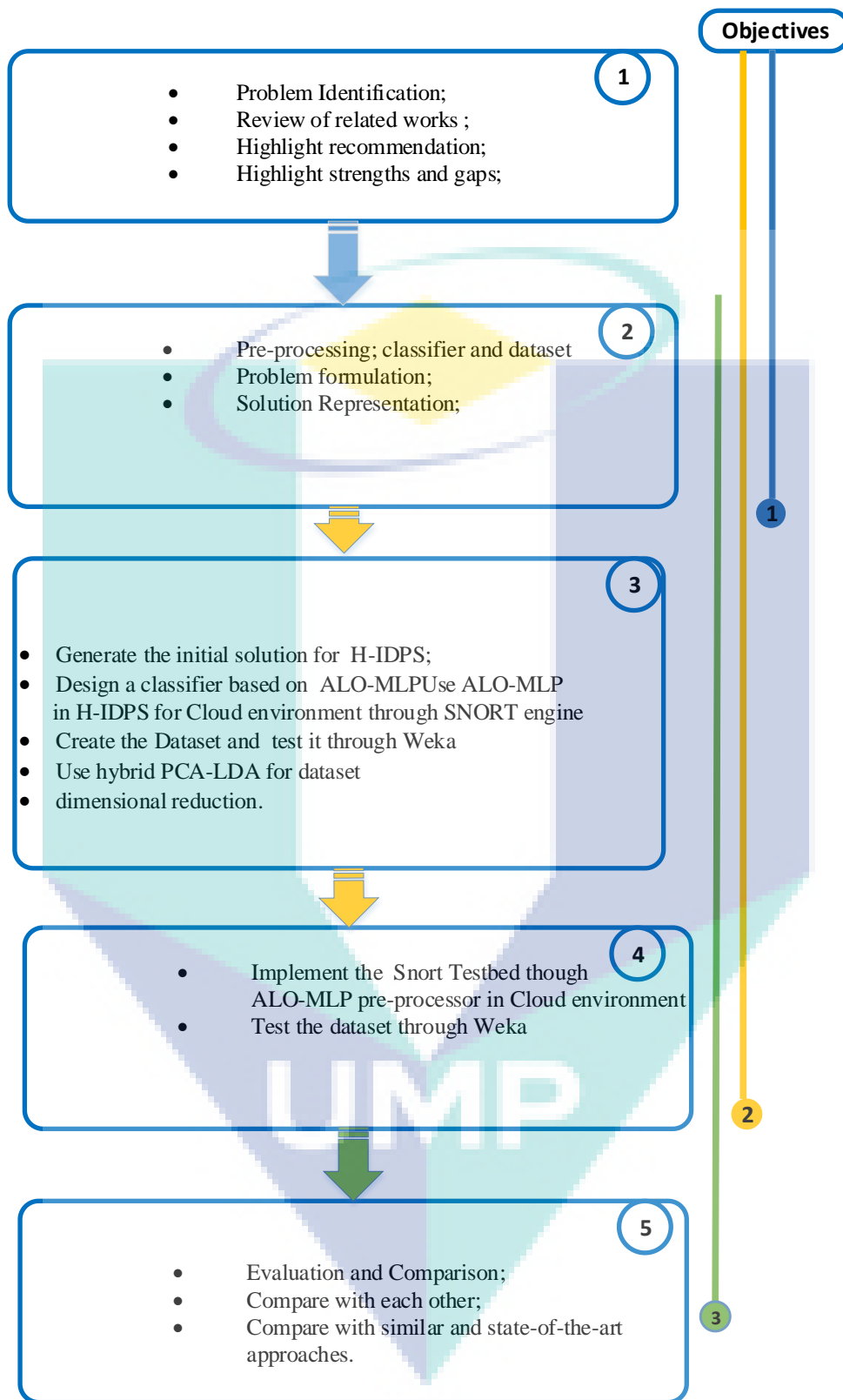


Figure 1.3 Operational research Framework.

## 1.6 Thesis Outline

The remainder of this thesis is organized in the following manner:

- **Chapter 2** discusses related studies in terms of their techniques, drawbacks, and challenges. It ends with a discussion on the history of the accomplished design.
- **Chapter 3** highlights the general requirements and considerations involved in designing a practical DDoS detection and prevention system, the algorithms used for attacks present in NSL-KDD, and the complexities of both the training and deployment phases.
- **Chapter 4** discusses the experiments conducted on the system for parameter measurements and evaluation through various metrics. The system is then compared with existing mechanisms.
- **Chapter 5** reports the results of our evaluations and comparative analyses.
- **Chapter 6** concludes the current research and offers suggestions for future work.

The logo for UIMP (University of Malakand) is a large, downward-pointing arrow shape. It is composed of four triangular sections meeting at a central point. The top-left and bottom-right sections are light blue, while the top-right and bottom-left sections are a darker, muted blue. The letters 'UIMP' are written in a bold, white, sans-serif font across the center of the arrow.

UIMP

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Overview

The present research aims to collect and investigate all the credible and effective studies to have examined the security of cloud computing. More specifically, the salient features and methods of previous papers will be extracted, and their characteristics described. To achieve these goals within the context of previous methods and assessment techniques, case studies covering new methods, datasets, and benchmarks are investigated with respect to the research questions raised in Chapter 1.

This Chapter provides an overview of previous research on cloud computing, DDoS, and H-IDPS. Section 2.2 gives a general background to cloud computing and its security issues. In Section 2.3, DDoS is critically reviewed to show how attacks influence the cybersecurity world, especially in cloud computing. Section 2.4 discusses the evacuation findings for existing H-IDPS in DDoS attacks on cloud computing to identify the current security challenges.

#### 2.2 Cloud Computing Security and DDoS Attack Classifiers

For decades, there have been fears surrounding privacy on the Internet, leading to agency computers functioning solely on isolated intranets connected via hard cables (da Silva Filho et al., 2018). Recently, the utilisation of the cloud has become ubiquitous—we store photos, emails, business files, and our very identities there—but many companies still fear the cloud, wondering how they can classify and secure their information if it has been entrusted to someone else. This concern has made cloud computing one of the more polarising issues for IT professionals.

According to the 2017 Cost of Data Breach Study: Global Overview (Ponemon Institute, June 2017), the average total cost of a data breach is US\$ 3.62 million, with the average cost for each lost or stolen record containing sensitive and confidential information being US\$141. While these costs decreased from 2016 to 2017, the numbers remain astronomical, particularly to small businesses who may be unable to recover from data breach liabilities. Not any industry is safe from cyberattacks, and the number of such attacks continues to grow.

Cyberattacks such as DDoS flooding have experienced extraordinary growth. They are often launched by sophisticated attackers—sometimes state-sponsored—that can overwhelm traditional and legacy security. Modern attackers are cyber spies that use traditional espionage tactics together with innovative and disruptive malware to bypass passive, defence-based security measures. To defeat such attacks, security must transform itself into an active agent that hunts today's attacks as aggressively as it classifies them.

To predict and defeat attacks in real time, cybersecurity must move to the cloud. The cloud can leverage big data and instant analytics across a large swath of end users, allowing known threats to be instantly addressed and those that seek to overwhelm security to be predicted. Cloud security must create a collaborative approach that analyses the event streams of normal and abnormal activity across all users to build a global threat-monitoring system.

Because many different users leverage the same cloud environment, cloud security is particularly suited to a collaborative environment that instantly predicts threats through a worldwide threat-monitoring system and shares them among all users under the cloud umbrella. Cyberattacks continue to disrupt our way of life with innovative new approaches to seeding malware and stealing our data. In turn, security must actively work to disrupt cyber spies, attackers, and terrorists through a collaborative security approach that leverages the big data and analytics that thrive within the cloud. The following sub-sections will review the basis of cloud computing and outline the challenges faced in detecting and preventing threats using machine learning classifiers.

## 2.2.1 Cloud Computing

The cloud refers to a distinct IT environment designed for remotely provisioning scalable and measured IT resources. The term originated as a metaphor for the Internet, which is a network of networks providing remote access to a set of decentralized IT resources. Before cloud computing became its own formalized IT industry sector, the symbol of a cloud was commonly used to represent the Internet in a variety of specifications and mainstream documentation of web-based architectures (Kaul et al., 2017). Figure 2.1 shows the importance of cloud computing in remote services and virtual desktop applications.

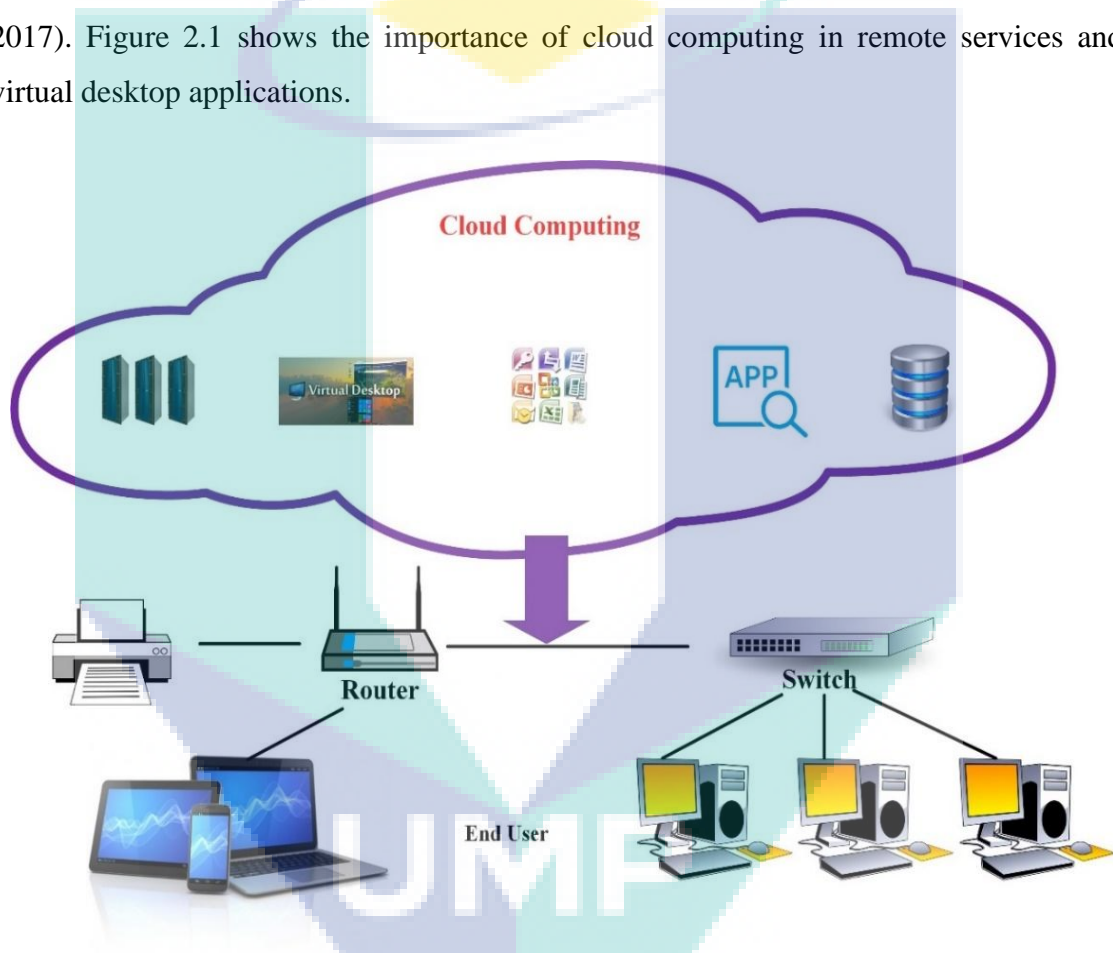
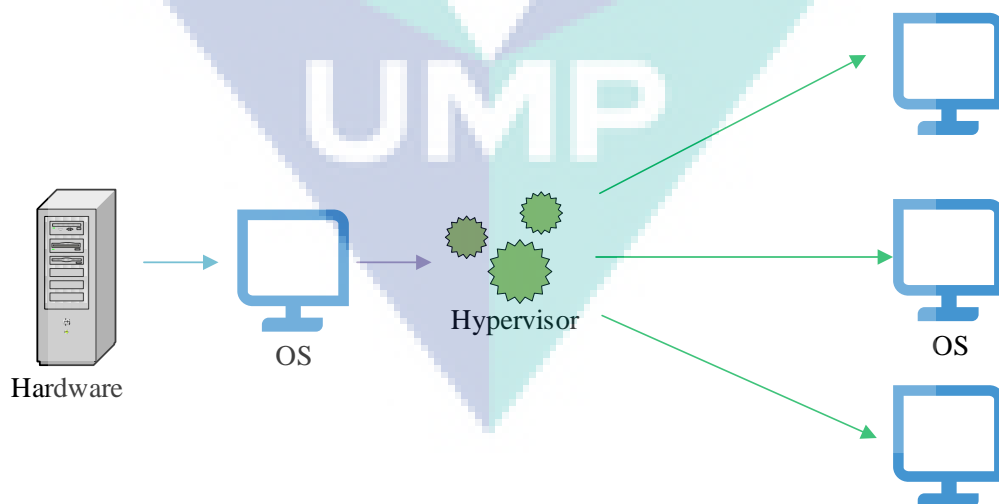
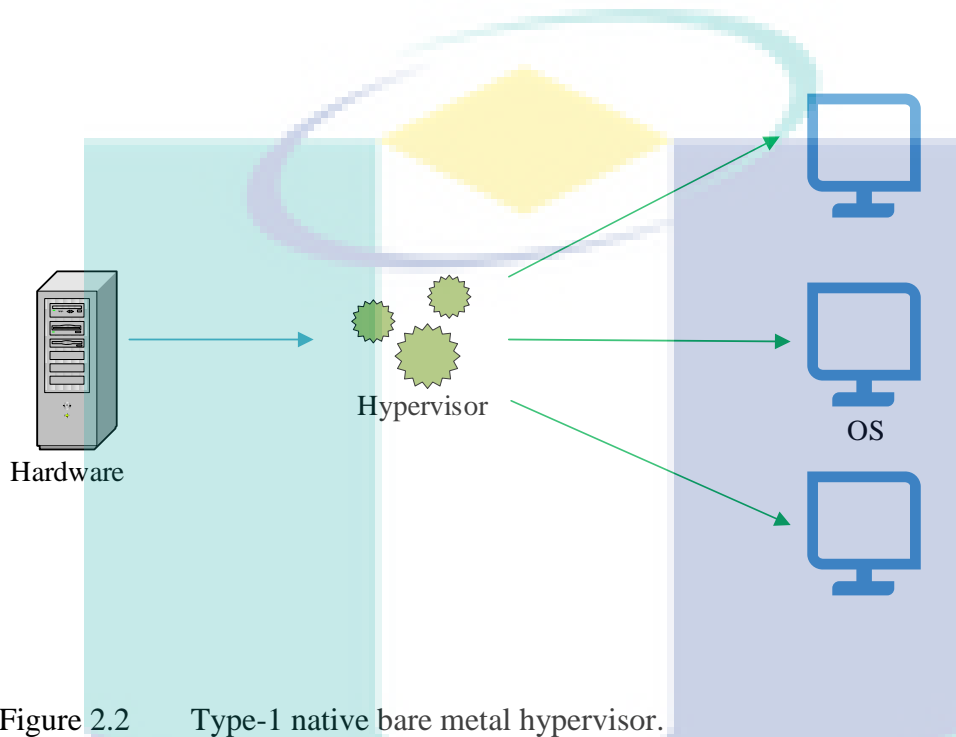


Figure 2.1 Cloud computing.

In the cloud environment, a hypervisor is an important component in the management of cloud servers, defined as the computer software, firmware, or hardware that creates and runs VMs (Perez-Botero et al., 2013). The hypervisor presents the guest operating system with a virtual operating platform and manages the execution of the guest operating system. Multiple instances of an operating system may share virtualized hardware resources.

Hypervisors can be classified into two types: Type-1 (native or bare metal) or Type-2 (hosted). These hypervisors run directly on the host hardware, controlling the hardware and managing the guest operating systems. Therefore, they are sometimes called bare metal hypervisors. Figures 2.2 and 2.3 illustrate the Type-1 and Type-2 hypervisors.





In conclusion, the hypervisor is primarily a management interface for the hardware primitives. The isolation of the central processing unit (CPU), memory, and input/output is now performed at the hardware level, with the hypervisor managing how much of the hardware resources can be used by a VM. With the ability to leverage these CPU extensions, the attack surface of the hypervisor shrinks considerably. Many security-related concerns about virtualization are unwarranted. Multiple hardware- and software-supported isolation techniques—as well as other robust security mechanisms such as access control and resource provisioning address the risks associated with these worries, especially DDoS attacks. In the following sections, we will explain how these technologies address security concerns.

## **2.2.2 Data Storage Security in Cloud Computing**

The recent rapid growth in the availability and popularity of cloud services has enabled convenient on-demand remote storage and computation. Security and privacy concerns, however, are preventing the wider adoption of cloud technologies. That is, in addition to the new security threats that emerge with the adoption of cloud technologies, a lack of direct control over one's data or computations requires new techniques to enhance a service provider's transparency and accountability.

### **2.2.2.1 Cloud Storage**

Cloud storage services offer remote maintenance, management, and back-up of data (More & Chaudhari, 2016). This is available to users over a network, typically the Internet, and allows files to be stored online so that the user can access them from any location via the network. Many of these services are free up to a certain number of gigabytes, with additional storage available for a monthly fee. All cloud storage services provide drag- and- drop access and the synchronization of folders and files between desktop/mobile devices and the cloud drive. They also allow account users to collaborate with each other when working on documents as shown in Table 2.1.

The public cloud is understandably synonymous with risk, as the end-user is not in control of the infrastructure. Although the CIO Mid-Year Review 2014 (a survey of CIOs in India) found that the number of executives citing security as the top concern dropped from 44% to 25% from 2013 to 2014 (Himmel & Grossman, 2014), cloud computing undoubtedly offers many possibilities for cybercriminals, not least of which is powerful DDoS attacks.

At the 2014 Black Hat conference, a pair of testers from Bishop Fox demonstrated how free-tier public cloud services could be pooled into a mini botnet that could mine the bitcoin cryptocurrency and potentially carry out DDoS attacks or password cracking. Targeted attacks such as Operation Ababil in 2013, which specifically focused on banking websites, have capitalized on Web vulnerabilities, and the number of such attacks may increase as more organizations supply and become dependent upon software-platform, and Infrastructure-as-a-Service (IaaS) (Gillman et al., 2015). As today, many personal cloud storage applications exist, for instance Apple iCloud, Microsoft OneDrive, Google Drive and Dropbox.

Dropbox is a cross platform application running on OS X, iOS, Android, Windows/Windows Phone and Linux. It offers a storage service, which can be accessed from the browser or automatically synced with the local file system via an ad-hoc client interface running in background. It also enables collaboration by allowing users to exchange up to-date contents located in a shared folder. The Web version of Dropbox offers a basic backup service, which can be used to revert to a prior version of a file, for instance to cope with accidental deletions or modifications (Boughorbel et al., 2017).

Table 2.1 Cloud storage advantages and disadvantages

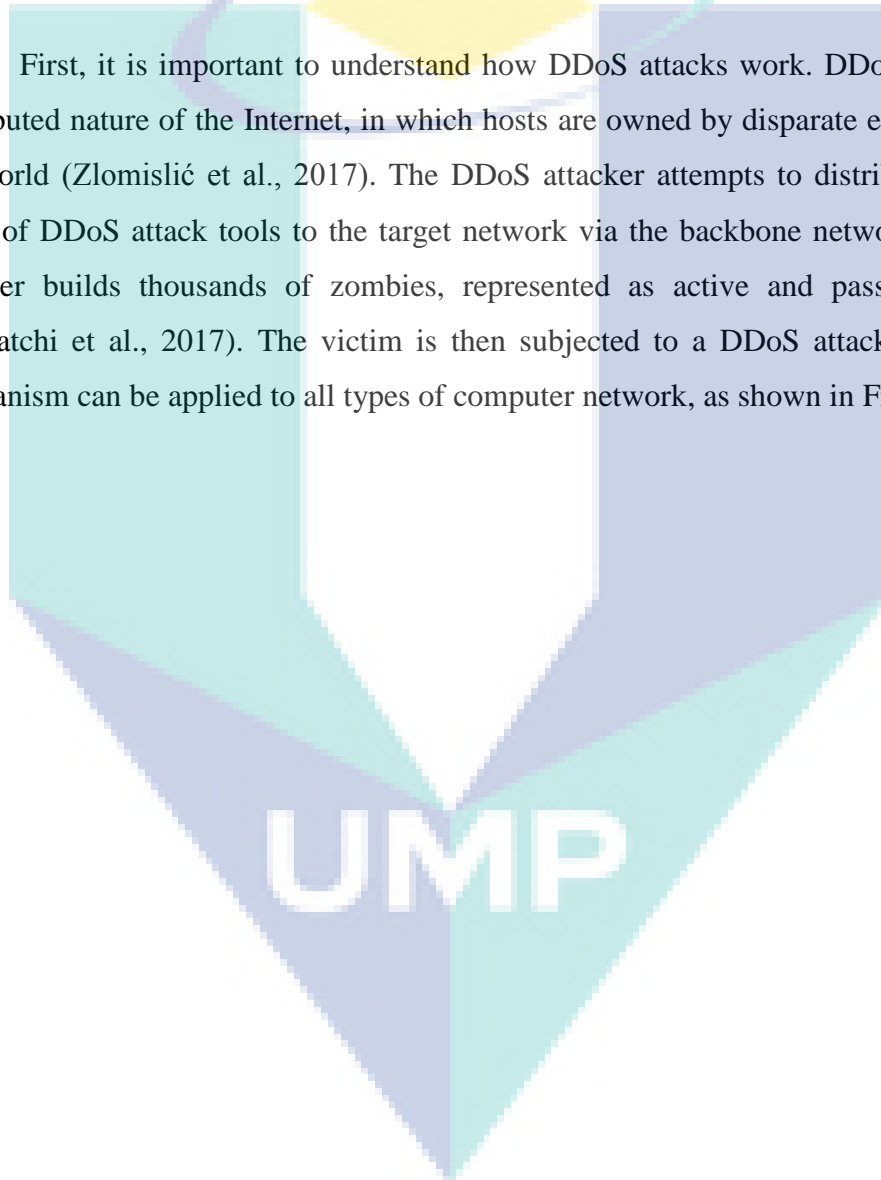
No.	Cloud storage	Description	Strengths	Weaknesses
1	Dropbox	Dropbox gives users the ability to share entire folders with other Dropbox account users, which allows updates to be viewable by all collaborators (Drago et al., 2012).	Ease of use. Very intuitive interface; for example, folders are shared by simply right-clicking the file or folder on the desktop and choosing to share.	Lowest amount of free storage of the offerings reviewed in this document. When inviting users to share files/folders, the email invitation must be sent to the email address associated with the users' Dropbox account (Sudharsan & Latha, 2013).
2	OneDrive	Can share content regardless of whether colleagues have accounts. Email notifications are sent when files are uploaded, downloaded, or added. Passwords can be set for important files.	Offers the most free storage of the options reviewed in this document. Like Google Drive, it is possible to edit documents within a browser, without having to open a client application like Microsoft Word.	Phishing email attacks have contained links to OneDrive (Daryabar et al., 2016).
3	Box	Box offers the ability to set time limits for user access to certain files. Further, it enables more control over user access to files and documents because security levels can be defined (Khan et al., 2018). Box is geared towards businesses and enterprises, but is also available for personal use	Storage of larger file sizes. Box is organized and user friendly, creating and organizing several layers of folders for all documents and data. It uses tagging to keep track of folders and files.	Flooding attack platforms.
4	Qualys	Qualys is a unique tool for securing personal file(s) in cloud computing (Everett, 2009).. As security threats continue to rise, regulatory compliance has mandated growing organizations to answer two crucial questions: 1) Are we secured from malicious threats and hackers? 2) Can we demonstrate regulatory compliance?	There is one advantage to the session-based authentication with v2.0: it triggers the reprocessing of the data.	Suffers from a performance overhead.

No.	Cloud storage	Description	Strengths	Weaknesses
5	White Hat Security	White Hat Security is an online security tool based on SaaS that protects other websites through commercial contract services (Denning et al., 2013).	Unlimited assessments, verification of every vulnerability, re-testing of every vulnerability on demand.	Weak design for risk management security tool solutions.
6	Okta	Okta is a replicate active directory for cloud applications with the same set of processor capabilities; however, Okta is a web service integration of salesforce.com intended for active directory integration and security applications (Smith, 2011).	Utilizes multifactor authentication for all SaaS/cloud apps via soft token, security questions or third parties that are fully integrated with Okta.	Okta is taking a cloud-based, on-demand approach to single sign-on, identity, and access (Usmani et al., 2018).
7	Proofpoint	One of the concerns in cloud data centre attacks is whether the cloud is secure (Mohamed et al., 2012). This issue can be answered by a proof point, which is a double-blind encryption technology that performs encryption on the day before the data leaves the customer's premises and is stored in the cloud.	Customers benefit from the scalability, performance, and economic advantages of Proofpoint's ongoing investment in the latest innovations in virtualization and cloud computing.	Centralized key management is often at the forefront of any encryption strategy, but good key management can be difficult to effectively implement and can prove difficult to fully understand. As such, any business needs to develop a detailed approach to create, store and remediate its keys.
8	Zscaler	Zscaler is a cloud security approach that can protect against advanced security threats (Ahmad et al., 2013). Forced policies for social media and cloud applications can prevent data loss without compromising user experience from any location and device, wherever they are in the world.	Zscaler is a SaaS platform, much like Salesforce.com, but is solely focused on security and compliance.	Lack of flexibility with respect to changes in security requirement tools.

### 2.2.3 Cloud DDoS Machine Learning Techniques for Classification of Attacks

Although the number of cloud projects has increased dramatically over recent years, ensuring the availability and security of project data, services, and resources is still a crucial and challenging research issue (Subashini & Kavitha, 2011). DDoS attacks are the most prevalent cybercrime after information theft. TCP and/or UDP flood attacks can exhaust the cloud's resources, consume most of its bandwidth, and damage an entire cloud project within a short period of time.

First, it is important to understand how DDoS attacks work. DDoS utilizes the distributed nature of the Internet, in which hosts are owned by disparate entities around the world (Zlomislíć et al., 2017). The DDoS attacker attempts to distribute different types of DDoS attack tools to the target network via the backbone network. Next, the attacker builds thousands of zombies, represented as active and passive attackers (Kamatchi et al., 2017). The victim is then subjected to a DDoS attack. This attack mechanism can be applied to all types of computer network, as shown in Figure 2.4.



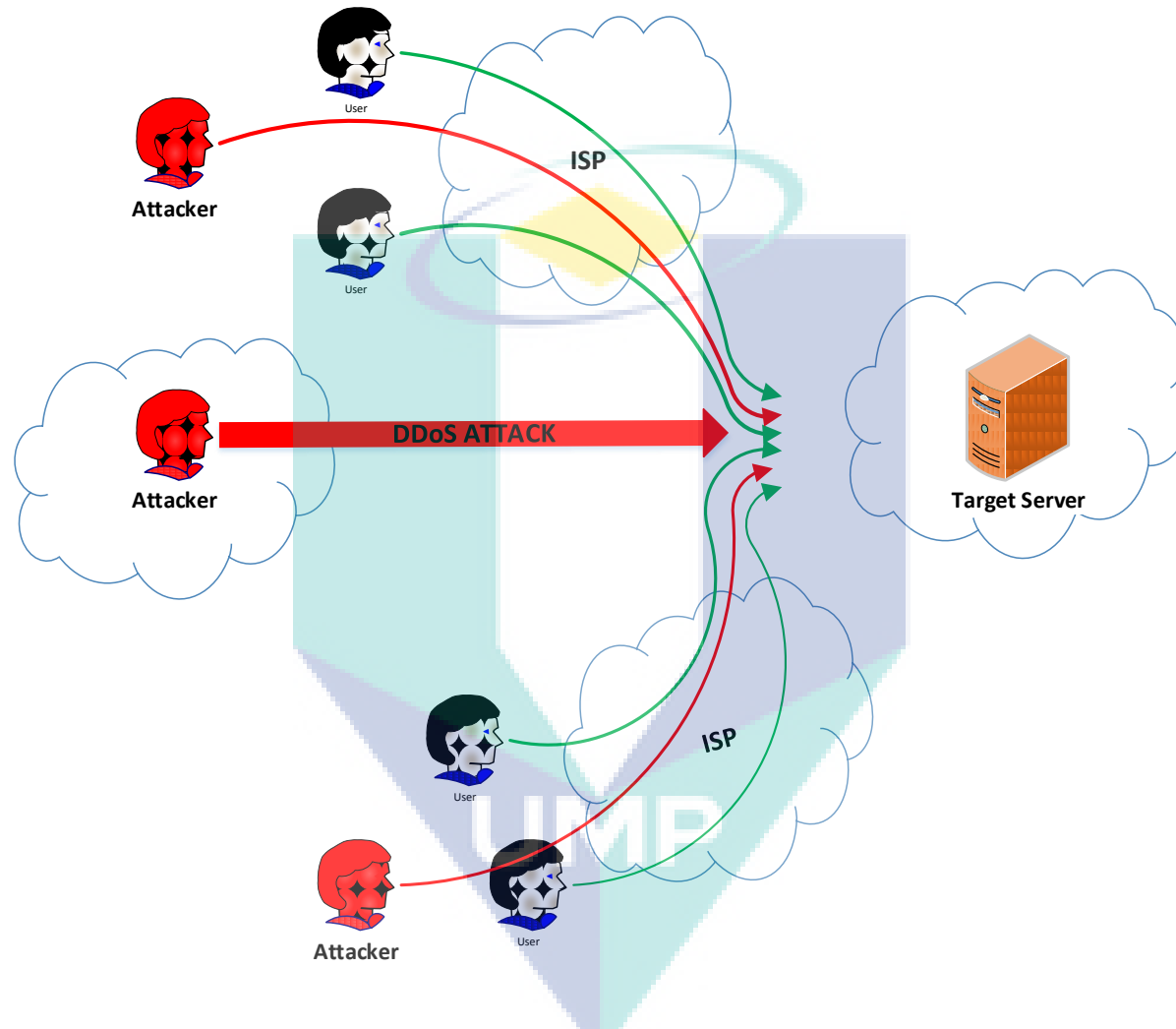


Figure 2.4 DDoS attack scenario in the cloud computing environment.

DDoS attack detection and prevention is a vital part of reactive DDoS mitigation (Bharot et al., 2017). There are two main methods for detecting attack traffic via IDPS, i.e. signature-based and anomaly-based techniques (Purwanto & Rahardjo, 2014). Signature-based detection consists of matching the packet signature against known attack signatures stored in a database. If the database is adequately populated, a low false negative rate is highly probable. However, this technique is unable to detect attacks that are not included in the database.

Although there are different vectors for DDoS attacks, they all aim to overwhelm servers, firewalls, or other perimeter-defined devices by sending large volumes of request packets (Saied et al., 2016). The network becomes overwhelmed to the point where a website becomes inaccessible. According to Black Lotus, the UDP flooding attack rate reached 53% in 2017 (Lotus, 2017). Moreover, the TCP and HTTP rates were 33% and 14%, respectively. Figure 2.5 illustrates the different types of DDoS attacks described by (Bhardwaj et al., 2016), and Table 2.2 describes various DDoS attack types.

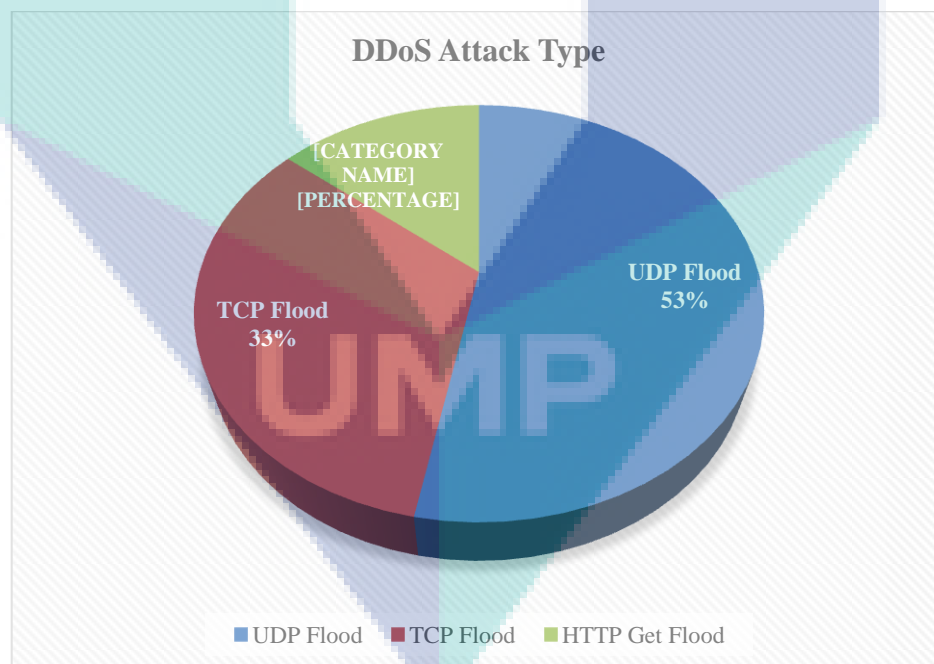


Figure 2.5 Black Lotus: three main DDoS flooding attacks.

Table 2.2 DDoS attack types

DDoS Attack	DDoS characteristics and types			
	Infrastructure	Application	Direct	Reflection
UDP flood	P	N/A	-	-
TCP flood	P	N/A	P	N/A
HTTP flood	P	P	P	N/A
ICMP flood	P	N/A	P	N/A
XML flood	P	P	P	N/A
Ping of death	P	P	P	N/A
Smurf	P	N/A	N/A	N/A

P = Partially N/A = Not applicable.

In cloud computing, the changing network traffic can bring about new DDoS attack types, which represent a serious risk to enterprise resources. Therefore, security administrators have a real need to employ efficient IDPS, especially H-IDPS. Such systems might be capable of learning from the network behaviour, by monitoring the characteristics of a single host and the events occurring within that host for suspicious activity. Figure 2.6 shows the characteristics of H-IDPS monitoring network traffic (only for that host), system logs, running processes, file access and modification, and system/application configuration changes.

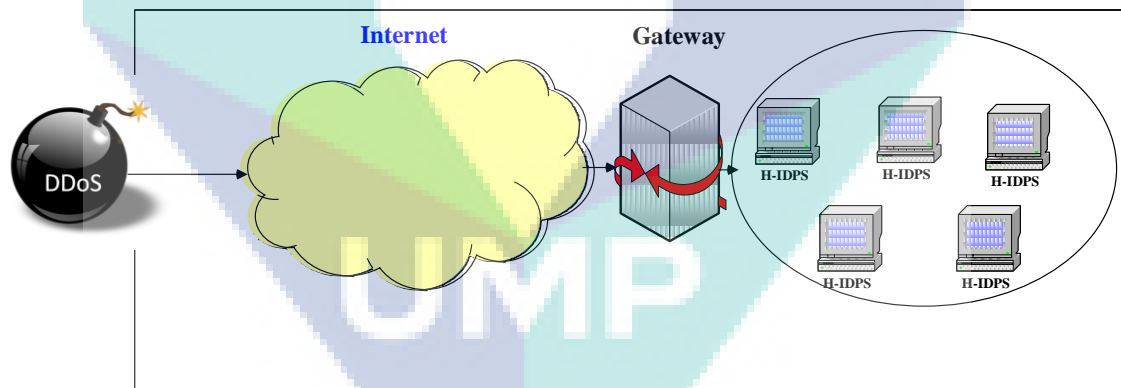


Figure 2.6 Schematic of H-IDPS.



Table 2.3 compares several detection methods based on criteria collected from existing surveys (Garcia-Teodoro et al., 2009; Nazer & Selvakumar, 2011).

Table 2.3 Comparison of H-IDPS methods based on collected criteria

H-IDPS technique	Alarm rate	Robustness	Resource consumption	Reliability	Speed
Signature	Low	Low	Low	Low	High
Anomaly	High	High	High	High	Moderate
Hybrid	Moderate	High	High	High	Moderate

Although H-IDPS will generally produce the correct classification, some events may be classified falsely. True positive (TP) and false negative (FN) classifications represent correctly classified events, whereas false positive (FP) and true negative (TN) represent wrongly classified events (Sahani et al., 2018). Recognizing a TN as being intrusive but not anomalous is a very difficult task that cannot be performed by the system itself. Instead, some human factor must be involved in the mechanism for recognizing such events. FPs that are not intrusive but anomalous are classified as intrusive but may be normal user events.

In general, to reduce the false alarm rate, an extra module known as a filter must be implemented before H-IDPS. In this way, false alarms can be eliminated from the output and the network administrator will only handle the relatively few alarms that are real intrusion attempts, thus saving time and manpower. This Chapter explains how the filter module works and how it reduces the number of false alarms. In conclusion, we must understand why machine learning is important in H-IDPS and identify the gaps in recent research; this is summarized in the next section.

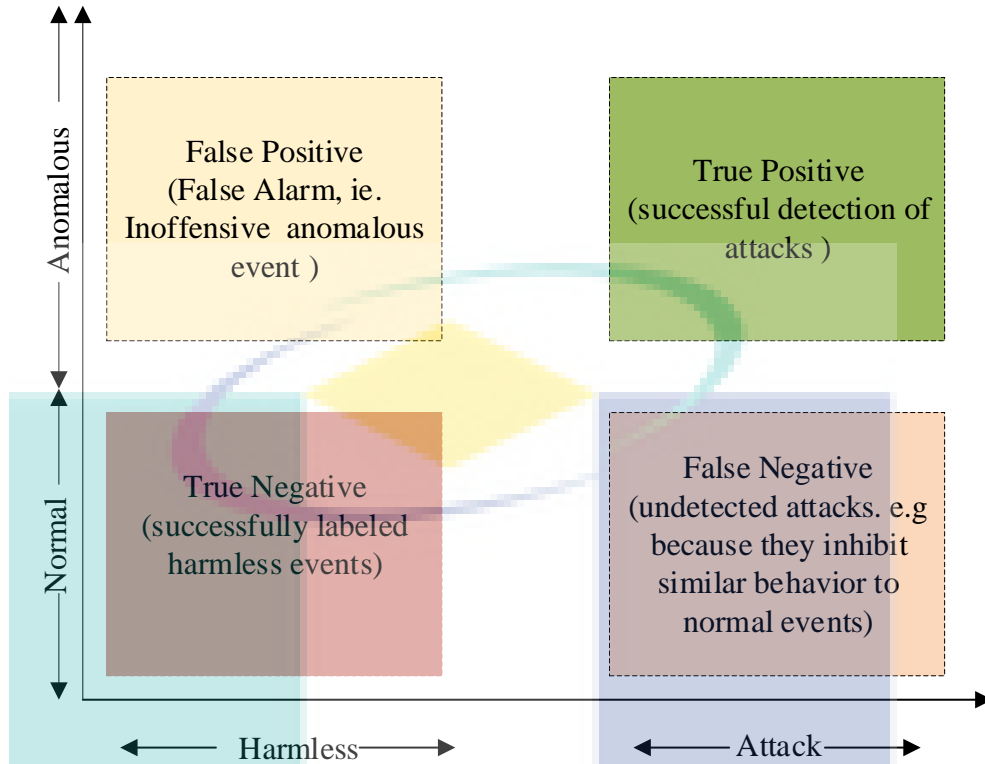


Figure 2.7 Classification of true/false negative/ positive.

Machine learning is the process of knowledge discovery from data without the need for explicit programming (Ndibwile et al., 2015). Machine learning techniques have become very popular over the past decade and are now used in many day-to-day applications such as image recognition, natural language processing, spam detection, intrusion detection and prevention, search engine applications, fault prediction, and stock market analysis (see Appendix A). In Table 2.4, shows several researchers have concentrated on machine learning approaches for detecting and preventing intrusion using fuzzy clustering, artificial neural networks (ANNs), support vector machines (SVMs), and fuzzy neural networks, which go beyond the conventional approach of generating results based on the detection rate and false negative rate.

Table 2.4 H-IDPS machine learning methods.

<b>IDPS technique</b>	<b>Characteristics/Advantages</b>	<b>Limitations/Challenges</b>
<b>Anomaly detection</b>	<ul style="list-style-type: none"> <li>• Uses statistical test on collected behaviour to identify intrusions.</li> <li>• Can reduce the rate of false alarms for unknown attacks.</li> </ul>	<ul style="list-style-type: none"> <li>• Requires a lot of time to identify attacks.</li> <li>• Detection accuracy is based on the amount of collected behaviour features.</li> </ul>
<b>IDPS based on fuzzy logic</b>	<ul style="list-style-type: none"> <li>• Used for quantitative features.</li> <li>• Provides better flexibility for uncertain problems.</li> </ul>	<ul style="list-style-type: none"> <li>• Has a lower detection accuracy than ANNs.</li> </ul>
<b>ANN-based IDPS</b>	<ul style="list-style-type: none"> <li>• Classifies unstructured network packets efficiently.</li> <li>• Efficiency of classification is increased when multiple hidden layers are used.</li> </ul>	<ul style="list-style-type: none"> <li>• Requires a lot of time and large number of training examples</li> <li>• Requires many samples to train effectively.</li> <li>• Relatively poor flexibility.</li> </ul>
<b>SVM-based IDPS</b>	<ul style="list-style-type: none"> <li>• Although sample data are limited, intrusions can still be correctly classified.</li> <li>• Manages many features.</li> </ul>	<ul style="list-style-type: none"> <li>• Classifies only discrete features. Therefore, there is a need for that feature to be pre-processed before application.</li> </ul>
<b>IDPS based on association rules</b>	<ul style="list-style-type: none"> <li>• Used to detect signatures of relevant known attacks in misuse detection.</li> </ul>	<ul style="list-style-type: none"> <li>• Not useful for unknown attacks.</li> <li>• Needs a lot of database scans to generate rules.</li> <li>• Can only be used for misuse detection.</li> </ul>
<b>GA-based IDPS</b>	<ul style="list-style-type: none"> <li>• Used to select best detection features.</li> <li>• High level of efficiency.</li> </ul>	<ul style="list-style-type: none"> <li>• Complex method.</li> <li>• Used in a specific way rather than general.</li> </ul>

The most common challenges faced by traditional methods are that IDPS generate false alarms and do not use proper standards or parameters to evaluate threats. This can lead to the misuse problem. We argue that it is necessary to test the robustness of machine learning mechanisms such as ANNs and MLP, especially in the diversified operating conditions prevalent in cloud scenarios.

### 2.2.3.1 ANNs

ANNs are information processing systems inspired by the behaviour of biological nervous systems (Modi & Acha, 2017). A neural network consists of many highly interconnected processing elements that work in unison to solve a specific problem. Each processing element is called a neuron, and these elements are controlled by an activation function (see Figure 2.8). The output of each neuron becomes the input to all neurons in the next layer. The learning process involves finding the best set of weights to solve a particular problem.

One of the design issues for ANNs is the type of transfer function used to compute the output of a node from its net activation. Popular transfer functions include:

1. Step function.
2. Signum function.
3. Sigmoid function.
4. Hyperbolic tangent function.

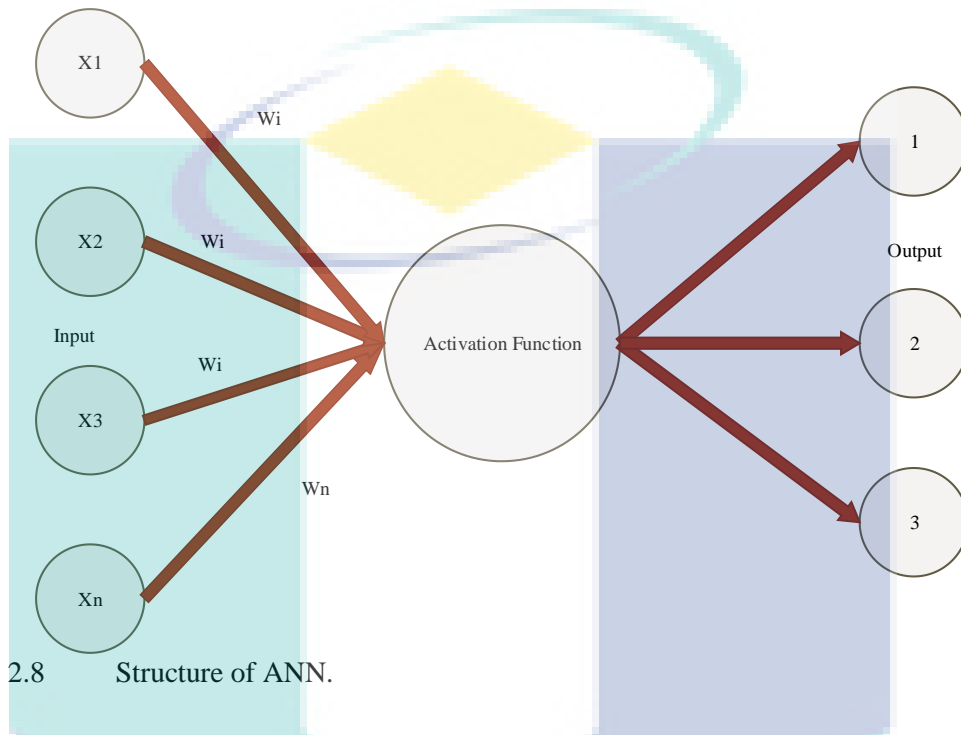


Figure 2.8 Structure of ANN.

The ANN gathers knowledge by detecting the patterns and relationships within the data fed into it and is essentially trained through experience. As a result, the ANN training time can be significant. (Shah & Trivedi, 2015) proposed a distributed system with an adaptive structure to detect attacks in the cloud platform. They used a KDD99 dataset in a testbed environment and achieved high accuracy with an acceptable computation time.

Chiba et al. (2016) proposed a cooperative hybrid network intrusion detection system (CH-NIDS) that detects network attacks in the cloud environment by monitoring network traffic while maintaining performance and service quality. In their NIDS framework, Snort signature-based detection is used to detect known attacks, with network anomalies identified using a Back-Propagation Neural Network (BPNN). By applying Snort first, the BPNN classifier can concentrate on detecting anomalies, thus reducing the detection time. To solve the BPNN problems of slow convergence and falling into local optima, they used a parameter optimization algorithm to ensure a high

detection rate, high accuracy, few false positives, and few false negatives at a reasonable computational cost.

### 2.2.3.2 MLP

To solve nonlinearly separable problems, it is possible to connect several neurons in layers to build an MLP. Each perceptron is used to identify small linearly separable sections of the inputs (Tang et al., 2016). The outputs of the perceptron are combined and fed into another perceptron to produce the final output. The hard-limiting (step) function used to produce the output prevents information on the real inputs from flowing into the inner neurons. To solve this problem, the step function can be replaced by the continuous *sigmoid function*. In an MLP, the neurons are arranged on an input layer, an output layer, and one or more hidden layers. The system applies neural projection architectures to detect anomalous situations. MLPs use advanced visualization features and provide an overview of network traffic. DDoS attacks in cloud computing can be input to an MLP for classification, as shown in Figure 2.9.

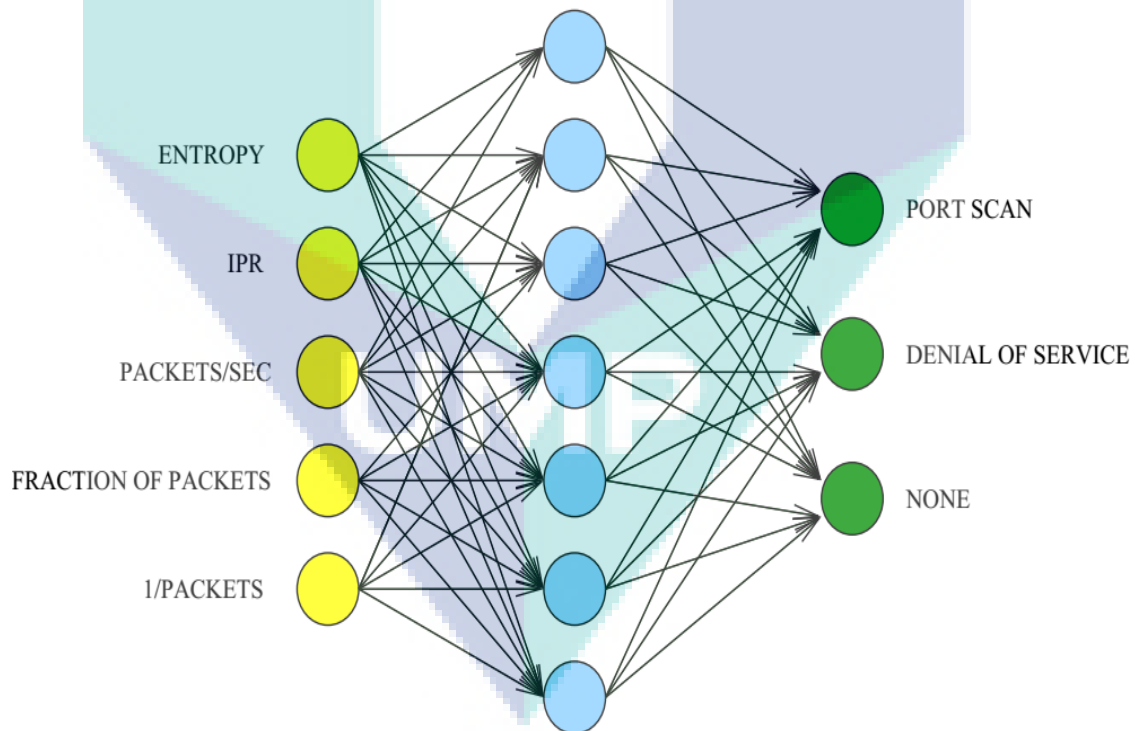


Figure 2.9 Schematic of MLP used in IDPS.

The learning rule for MLPs is known as the generalised delta rule or the backpropagation rule. The generalised delta rule calculates an error function for each input and back-propagates the error from one layer to the previous one. The weights for

each node are adjusted in direct proportion to the error. In this algorithm, only the winning weight vector can change its value after each iteration. The other weight vectors remain unchanged. (Mukhopadhyay et al., 2011) used the backpropagation neural network approach, in which the error is propagated from the output layer to the hidden layer and then to the input layer. Corchado & Herrero (2011) used a system called mobile visualization connectionist IDS and IPS (MOVCIDS).

### **2.2.3.3 K-Nearest Neighbours**

The K-Nearest Neighbours (KNN) approach is a simple technique for classifying data. It computes the distance between two points, and then classifies unlabelled data accordingly. Deshpande et al. (2014) proposed an H-IDS based on analysing the failed system calls trace and classifying it using KNN. This would reduce the computational burden, detect the intrusion early, and alert the user to the threat.

Their proposed scheme provided security at the infrastructure layer, where each VM used the IDS. The proposed mechanism achieved 96% average intrusion detection sensitivity. However, this system had only a limited view of the virtual network activity, so it was only able to detect malicious activity on the machine where it occurred.

Ghosh and Mitra (2015) presented a hybrid KNN and Neural Network KNN and NN algorithm to improve the classification performance. They used rough set theory and the idea of information gain to select 25 features from the NSL-KDD dataset; the results indicate a reduction in both training time and memory usage. KNN was used to classify normal and abnormal data, and then the abnormal class was passed to the NN to classify specific attack types such as DOS, U2R, R2L, and Prob. The results emphasized that, for NSL-KDD, information gain was more suitable than rough set theory for choosing appropriate features, and the proposed KNN-NN hybrid multilevel classification increased the accuracy of the intrusion detection system. Moreover, the proposed KNN-NN hybrid multilevel classification offered an improvement over KNN and NN, achieving 76.54% accuracy. Nevertheless, there are some limitations to their approach because of the very high level (23.46%) of false alarms.

Bhat et al. (2013) introduced a machine learning method that constructs IDS on the VM monitor (i.e. hypervisor). First, they used a naïve Bayes (NB) tree to classify

packets based on the NSL-KDD training dataset. This part contributes to a better classification mechanism by determining the most important features. Second, they used a hybrid of an NB tree and a random forest (RF) to predict the class of data based on the similarity of connection features. They compared the proposed method with many IDSs using machine learning. The results suggest that the hybrid NB tree and RF outperforms the NB tree in terms of accuracy and the false negative rate.

#### **2.2.3.4 Fuzzy Logic**

Fuzzy Logic (FL) provides a simple way of arriving at a definite conclusion based upon vague, ambiguous, noisy, imprecise, or missing input information (Sanchez et al., 2017). The process of reaching this definite conclusion is as follows: (i) All input values are fuzzified into fuzzy membership functions. (ii) Fuzzy rules are generated in the form of IF-THEN statements. (iii) Given an instance, some of the fuzzy rules will be activated. (iv) The activated rules are combined in the rule base to compute the fuzzy output distribution. (v) The fuzzy output distribution is defuzzified to obtain a crisp output value.

Mkuzangwe and Nelwamondo (2017) proposed an FL network IDS to detect Neptune, which is a type of TCP SYN flooding attack. The NSL-KDD dataset was used to train and evaluate the system, which was compared with a decision tree. The results indicate that the performance difference, in terms of predicting the proportion of Neptune cases in the test data, between the proposed system and the decision tree is negligible.

FL can be applied in IDS when some features are considered as fuzzy variables. Iyengar et al. (2014) presented a new defence mechanism to mitigate DDoS traffic in cloud data centres. They used FL to define rules based on a predefined traffic pattern, enabling their proposed mechanism to infer the traffic class (normal or attack) based upon acquired knowledge. Experiments carried out in a simulated environment attained a classification accuracy of 86.93%.

#### **2.2.3.5 Evolutionary Computation**

Evolutionary Computation (EC) refers to optimization algorithms which are inspired by biological evolution, such as Genetic Algorithms (GAs), Particle Swarm



Optimization (PSO), ant colony algorithms, and whale optimization (Ghamisi & Benediktsson, 2015). These methods are global heuristic search techniques that select network features or determine the optimal parameters for use in other techniques, thus improving the performance of the IDS. For example, Raja and Ramaiah (2016) presented a hybrid feature selection and multiclass classification algorithm to detect attacks in VMs. The authors proposed a security mechanism integrating a GA with discrete PSO to select the best features from the NSL-KDD dataset. They then integrated a hidden NB approach into an intelligent agent-based multi-class SVM. The performance results show that their hybrid algorithms can achieve an accuracy rate of greater than 95%.

#### **2.2.3.6 Probabilistic Reasoning**

Probabilistic Reasoning (PR) combines probability theory with deductive logic (i.e. reasoning from one or more statements) to deal with uncertain data. Most PR methods used in IDS rely on Dempster–Shafer, Markov, and entropy theory. As described by Lonea et al. (2013), Dempster–Shafer three-valued logic and fault trees can be used to analyse and detect DDoS attacks in the cloud environment. The proposed solution combines Snort IDS in each VM with a cloud fusion unit (CFU) at the front end. The alerts from all VMs are stored in a MySQL database within the CFU, and then converted to basic probabilities which are used to detect attacks. This method deals with uncertain states to reduce the false negative rate and meet the detection rate and computation time requirements.

Using the Hidden Markov Method (HMM), Chen et al. (2012) detected the sequence and frequency of attacks. They collected multiple logs from a campus network and normalized them to produce a uniform format. The features of the data were then extracted and mapped to observed actions and events, before being passed to the trained HMM to distinguish whether the attack exists. Several related work summarizes the existing classifiers techniques as shown in Table 2.5.



Table 2.5 Summary of classification techniques.

Author (s)	Technique	Type of intrusion based on Alert Analysis	Type of IDS based on source of data	Type of IDS based on position	Dataset	Results
Deshpande et al. (2014)	KNN	Anomaly	Host	VM	System call traces (real data captured)	Sensitivity (96%)
Ghosh and Mitra (2015)	Information Gain (IG) with KNN & ANN Rough set with KNN & ANN	Anomaly	Network	N/A	NSL-KDD	IG with KNN & ANN (76.54%) Rough with KNN & ANN (74.59%).
Bhat et al. (2013)	NB tree; RF & NB tree	Anomaly	Hypervisor	Hypervisor	NSL-KDD	NB tree (99.65%) NB tree & RF (99.1%)
Iyengar et al. (2014)	FL	Anomaly	Network	Back End	Simulated traffic	Accuracy (86.93%)
Manickam and Rajagopalan (2018)	Type-2 fuzzy & k-means & GA & fuzzy NN.	Anomaly	Network	N/A	CIDD	Detection rate (98.598%)

For the purposes of this review, we have considered network intrusion simulations based on the NSL-KDD dataset. Thus, it is prudent to explore the results of recent IDS construction activities for that dataset. In addition to ensemble approaches, many machine learning techniques have been applied to IDS development. Some of the most popular approaches are hybrid methods, where a classification task is usually decomposed into feature selection or reduction and the classification of pre-processed data. The chief advantage of this approach is the significant decrease in computational cost, and many lightweight IDSs have been built along these lines. Additionally, favourable classification results have ensured that hybrid IDS construction approaches remain an active research area.

Hota and Shrivastava (2014) made a comparative study of various hybrid approaches for both binary (normal vs. attack) and multi-class classifications of the NSL-KDD dataset. Each hybrid implementation used the information gain (IG) feature selection and one of five classification algorithms: MLP, C4.5, RF, and REP tree. The authors reported that the best performance was achieved with an IG-RF hybrid classifier.

Pervez and Farid (2014) defined a hybrid approach based on feature selection and subsequent classification using the NSL-KDD dataset. Feature selection was implemented following the Leave-One-Out (LOO) method, and, as a classifier, the authors deployed SVMs in a One-against-the-Rest Multi-Class Configuration (OAR-SVM). Their experiment showed that the greatest classification accuracy was achieved by evaluating 14 selected features.

Enache and Patriciu (2014) developed a two-stage hybrid approach: (i) feature selection with an IG algorithm and (ii) classification with an SVM method for binary (normal vs. attack) IDS classification. In addition, the authors chose to introduce a meta-optimization based on swarm intelligence algorithms to find the optimal set of classification parameters for the SVM. Two approaches were used to optimize the SVM classification parameters: PSO and Artificial Bee Colony (ABC). The experimental results for the NSL-KDD dataset indicated that an ABC-SVM approach achieved slightly higher precision than PSO-SVM.

Eid et al. (2011) proposed a simple hybrid classifier as a solution to the IDS classification problem. A GA was implemented as a wrapper method for feature selection, in conjunction with an NB classifier. The optimal subset of features was found by minimizing the classification error of the NB classifier trained with a given subset of features. In addition to feature selection, the authors implemented the Entropy Minimization Discretization (EMD) method to discretize the input data. The method was applied to the NSL-KDD dataset, with the whole set used for training, and the effectiveness of the proposed method was evaluated using 10-fold cross-validation.

Emiro De la Hoz et al. (2014) implemented a two-component hybrid approach, with a feature selection and classification stage. They employed multi-objective feature selection, with the non-dominated Sorting Genetic Algorithm (NSGA) implemented to find the subset of features that maximized the Jaccard coefficient for each class in the dataset. The NSL-KDD dataset was classified by the growing hierarchical self-organizing maps (GHSOM). Similar to Eid et al. (2011), the whole NSL-KDD dataset was used in the training phase, and the results were based on 10-fold cross-validation with a reported accuracy of 95.60%.

Rastegari et al. (2015) developed an IDS based on GA optimization. Binary classification (normal vs. attack) of the NSL-KDD dataset was performed using a set of IF-THEN rules applied to the selected features. The features for rule construction and condition boundaries were selected by GA optimization, with the goal of minimizing the number of misclassified instances. Additionally, the authors implemented Correlation-Based Feature Selection (CFS), the Consistency Subset Evaluator (CSE), and the selection of only real-valued features. Their results indicate that the developed approach is comparable to other single-stage learning methods.

Alpha profiling was applied to the whole NSL-KDD dataset to combine the protocol and service features into a single “alpha” feature. To reduce the training time, beta profiling was deployed to remove redundant training pairs from the training set. Feature selection was based on three approaches: Filtered Subset Evaluation (FSE), CFS, and CSE. The authors reported that their Alpha FST Beta OSLEM approach could reduce both the dimensionality and training set size without compromising the classification accuracy.

Kanakarajan and Muniasamy (2016) presented an approach based on a greedy randomized adaptive search procedure with Annealed Randomness (GAR-forest) classifier for both binary (normal vs. attack) and multi-label classification of NSL-KDD. The GAR-forest approach uses the meta-heuristic Greedy Randomized Adaptive Search Procedure (GRASP), which generates a set of randomized adaptive decision trees. Feature selection was implemented through IG, Symmetrical Uncertainty (SU), and CFS. The authors reported that the GAR-forest classifier outperformed RF, C4.5, NB, and MLP classifiers. Their feature selection method also resulted in improved classification accuracy.

Hassanien et al. (2014) presented a multi-layer IDS based on three stages: (i) feature extraction through Principal Component Analysis (PCA), (ii) binary (normal vs. anomalous) classification with a GA, and (iii) multi-class categorization of anomalous instances with decision trees. The GA classification was performed as a set of IF-THEN rules, with each observation labelled as either normal network traffic or a network intrusion. The experimental procedure was conducted on the NSL-KDD dataset. An analysis of the developed approach found that two-layer classification offered more reliable classification results than single-stage classifiers. A similar approach was developed by (Pajouh et al., 2017).

As a feature reduction method, they implemented a Linear Discriminant Analysis (LDA) algorithm. The first-tier, binary (normal vs. anomalous) classification was performed with an NB classifier, and anomalous data were then classified more precisely in the second tier using  $k$ NNCF ( $k$ NN with a certainty factor).

The analysis of Hassanien et al. (2014) and Pajouh et al. (2017) indicates that the latter managed to obtain considerably better classification results.

Table 2.6 provides an overview of popular IDS classification approaches based on the NSL-KDD dataset.

Table 2.6 Popular NSL-KDD classification approaches based on feature selection and classifier method.

Reference	Feature selection/Pre-processing	Classification method
Eid et al. (2011)	GA and EMD	NB
Hassanien et al., 2014	PCA	GA-DT
Enache and Patriciu (2014)	IG	PSO-SVM
Hota and Shrivastava (2014)	IG	MLP
Emiro De la Hoz et al. (2014)	NSGA	GHSOM
Pervez and Farid (2014)	LOO	OAR-SVM
Pajouh et al. (2017)	LDA	NB-kNNCF
Rastegari et al. (2015)	CFS	GA classifier
Kanakarajan and Muniasamy (2016)	IG	GAR-forest

### 2.3 DDoS Benchmark Dataset for Verification of Machine Learning Classifiers

Recently, machine learning-based methods for security applications have been gaining popularity as machine learning techniques become more advanced (Kim et al., 2018). However, the major challenge with these methods is to obtain real-time and unbiased datasets. Many benchmark datasets cannot be shared because of privacy issues, or lack certain statistical characteristics. Because of this, researchers prefer to generate datasets for training and testing purposes in simulated, or closed experimental environments, which may lack comprehensiveness. Machine learning mechanisms trained with such datasets generally result in a semantic gap between the test results and their application. There is a lack of research on the effectiveness of these mechanisms when applied across multiple datasets obtained from different environments.

Various security research groups have introduced network intrusion datasets to assess various intrusion detection methods and unknown attacks (Behal & Kumar, 2016). These datasets can be classified into three categories: public datasets, private datasets, and network simulation datasets (J. Singh et al., 2012). Several tools have been used to generate most of the public and private intrusion datasets.

### 2.3.1 NSL-KDD

The NSL-KDD dataset is a heterogeneous dataset consisting of 41 features and a class variable. The data cover both discrete and continuous values, with some of the discrete values being symbolic (Ingre & Yadav, 2015). In the KDD Cup 1999 dataset, several instances are redundant and exhibit potentially biasing learning mechanisms towards frequent records. To solve this problem, the NSL-KDD dataset retains only one of the duplicated records (Tavallaee et al., 2009).

Hatef et al. (2018) presented a comprehensive and accurate solution that can detect and prevent intrusions in cloud computing systems using a hybrid method. They also applied their technique to the NSL-KD99 dataset. Although they evaluated their technique in terms of the accuracy, reliability, and availability of false alarms, they did not report the instances of correct and incorrect classifications.

Osanaiye et al. (2016) used an ensemble-based multi-filter feature selection method that combines the output of four filtering techniques to achieve the optimum selection. They performed extensive experimental evaluations using an intrusion detection benchmark dataset, NSL-KDD, and a decision tree classifier. Their method can effectively reduce the number of features from 41 to 13 while achieving a high detection rate and classification accuracy. However, the dataset reduction in this approach means that not all the dataset features are used.

### 2.3.2 DARPA Family

The US Defence Advanced Research Projects Agency (DARPA) is responsible for the development of emerging technologies in the military sector (Maher et al., 2014). DARPA-sponsored datasets are all synthetically generated, and questions have been raised about the realism of the underlying traffic mechanisms. In addition, none of the DARPA datasets was recorded on a network connected to the Internet. They usually contain a large degree of anomalous traffic that is not caused by malicious behaviour.

### 2.3.3 CAIDA

This dataset contains approximately one hour of anonymized traffic traces from a DDoS attack on August 4, 2007. The one-hour trace is split into various packet capture (PCAP) files (Grossman et al., 2009). The compressed dataset has a total size of

5.3 GB (21 GB uncompressed), and includes only attack traffic to the victim and responses to the attack from the victim (K. Singh et al., 2017). The payload has been removed from all packets. These traces can be read with any software that can handle the PCAP TCPDUMP format, such as the Coral Reef Software Suite, TCP DUMP, Wireshark, and many others.

Jiao et al. (2017) proposed a real-time TCP-based DDoS detection approach which extracts the effective features of TCP traffic and distinguishes malicious traffic from normal traffic using two decision tree classifiers. They evaluated their approach using CAIDA and achieved an attack detection rate of more than 99% with a false alarm rate of less than 1% in a cloud computing environment. One main issue with this study is that it is based on only one DDoS TCP flooding attack, and so it should also be applied to a UDP flooding attack.

Karimi et al. (2016) developed a distributed architecture-based IDS that can detect network anomalies in real time. However, they only divided the datasets into two groups of attacks and did not consider applying data reduction to the CAIDA dataset. However, these cloud/DDoS IDPS datasets underwent ML-based pre-processing, one part of which is dimensional reduction. The two most popular algorithms for dimension reduction are PCA and LDA.

### **2.3.3.1 Dataset Dimension Reduction**

Dimensionality reduction is a field of machine learning in which high-dimensional data are mapped to a lower dimension while preserving important features of the original dataset (Cunningham & Ghahramani, 2015). Two well-known dimensionality reduction techniques are PCA and LDA. Although many prior studies have developed feature selection and feature extraction techniques to reduce the size of the data under consideration, none has focused on determining by how much the dataset should be reduced.

PCA has been widely used to extract the most relevant information from a dataset. It has been successfully used in face recognition applications, where PCA is employed to derive a new set of uncorrelated features from a set of correlated ones. Thus, PCA generates a set of orthogonal basis vectors, allowing the data to be expressed



as a linear combination of that basis. Some researchers have claimed that PCA introduces some classification task problems, as more processing is required whenever new data are added, and the data reduction is not invariant under certain transformations. Using PCA in the design of an IDPS will reduce the complexity of the system whilst achieving higher classification accuracy. The process of PCA can be described as follows:

1. Compute mean vectors for the input features dataset ( $x_i$ )

$$\text{Mean } \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

2.1

2. Calculate the scatter matrix as the covariance matrix

$$S = \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T$$

2.2

3. Compute eigenvectors and eigenvalues
4. Sort the eigenvectors in descending order to give the vector  $W$
5. Project the principal components onto the input features dataset

$$Y = W^T x$$

2.3

UMP



Table 2.7 Summary of PCA approaches.

Approaches	Advantages	Disadvantages
<b>Reddy et al. (2017)</b>	IDPS with the union of most efficient features selected by PCA can reduce the computational complexity of the system. Along these lines, an improved version of K-means clustering was developed for enhanced classification accuracy.	PCA has been utilized for feature extraction, whereby components are fundamentally anticipated into a principal space and at that point elements are chosen based on the eigenvalues. However, the elements with the largest eigenvalues might not provide the classifier with the ideal affectability.
<b>Keerthi Vasan and Surendiran (2016)</b>	PCA experiments were conducted using various classifier algorithms on two benchmark datasets, namely KDD CUP and UNB ISCX. The results show that the first 10 principal components are effective for classification. The classification accuracy with 10 principal components is above 99%.	The original 41 features (KDD) and 28 features (ISCX) were used without any dataset normalization.
<b>Eduardo De la Hoz et al. (2015)</b>	This approach hybridizes statistical techniques and self-organizing maps for network anomaly detection. PCA and Fisher's Discriminant Ratio (FDR) were considered for feature selection and noise removal.	Further investigation is required to determine how fast IDS implementations will need to be to cope with current link bandwidths.
<b>Thaseen and Kumar (2014)</b>	Proposes a novel method of integrating PCA and SVM by optimizing the kernel parameters using an automatic parameter selection technique. This technique reduces the training and testing time needed to identify intrusions, thereby improving the accuracy. Tested on the KDD dataset.	Minimal resources are consumed as the classifier input requires reduced feature set, thereby minimizing the training and testing overhead.
<b>Lee et al. (2013)</b>	Proposed an online Over-Sampling PCA (OSPCA) algorithm.	Most anomaly detection methods are typically implemented in batch mode, and thus cannot be easily extended to large-scale problems without sacrificing computation and memory requirements.

Another widely used feature extraction approach is LDA, which is based on the within-class scatter and between-class scatter. To separate different classes, the between-class scatter must be maximized, and the within-class scatter must be minimized. The LDA algorithm proceeds as follows:

1. Compute mean vectors for the input features dataset ( $x_i$ )

$$\text{Mean } \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad 2.4$$

2. Calculate the within-class ( $S_w$ ) and between-class ( $S_B$ ) scatter matrices

$$S_w = \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T \quad 2.5$$

$$S_B = \sum_{i=1}^n N_i (\bar{x} - m)(\bar{x} - m)^T \quad 2.6$$

3. Find linear discriminants by computing the eigenvalues for  $S_w^{-1} S_B$
4. Select the linear discriminants for the new feature set by sorting and choosing the eigenvectors  $W$  with the highest eigenvalues
5. The feature set obtained by the linear discriminants is then used to obtain the transformed input dataset according to

$$Y = X.W \quad 2.7$$

Table 2.8 Summary of LDA approaches.

Approaches	Advantages	Disadvantages
Elkhadir et al. (2017)	Proposed an improved median nearest neighbours LDA (median NN-LDA), which performs well without satisfying the above two conditions. Their approach can effectively determine the local structure of data by working with samples that are near to the median of every data class.	There are concerns about the nature of the data class distribution.
Aburomman and Reaz (2016)	Developed an efficient IDS in which an ensemble of LDA feature extraction algorithms were implemented.	Feature extraction addresses the problem of finding the most compact and informative set of features.
Saad et al. (2015)	Proposed an IDS implementation that achieves a detection accuracy rate of 97.75%.	Still need to develop more accurate algorithms combining D-LDA with some other classification methods for intrusion detection.

## 2.4 Evaluating Findings for Existing H-IDPS Cloud DDoS Attack Classifiers

To compare all previous approaches on an equal footing, our examination was restricted to the overall classification accuracy based on the same type and size of dataset. Only studies that applied the full NSL-KDD dataset were used for comparison (see Table 2.9).

Table 2.9 Comparison of studies that classified the NSL-KDD dataset in terms of overall accuracy

<b>Authors</b>	<b>Approach</b>	<b>Accuracy</b>
Pajouh et al. (2017)	IG-GAR	82.00%
Hassanien et al. (2014)	PCA-BFtree	68.28%
Kanakarajan and Muniasamy (2016)	LDA-NB-kNNCF	78.90%
Pervez and Farid (2014)	LOO-OAR-SVM	82.68%
Tavallae et al. (2009)	MLP	77.41%

## 2.5 Chapter Summary

The distributed and open structure of cloud computing and related services has become an attractive target for potential cyberattacks by intruders. H-IDPS are largely inefficient when deployed in cloud computing environments because of their openness and specific characteristics. As with any developing system, IDPS in cloud computing must be improved. This Chapter has discussed H-IDPS in terms of the threats it is intended to catch, the challenges it faces, and the types of alerts that it triggers. In addition, the detailed in H-IDPS has been reviewed, as this is the basis on which the research presented in this thesis is built. In summary, current H-IDPS approaches lack the ability to identify and distinguish TP, TN alerts and accuracy during cloud DDoS attacks. The next Chapter describes a methodology that can overcome this issue.

## CHAPTER 3

### METHODOLOGY

#### 3.1 Overview

This Chapter describes the design of an H-IDPS mechanism for a cloud computing environment. Section, 3.2 shows how the research objectives can be achieved within this methodology. Each individual process is separately explained in the following subsections. The main mechanism is used to detect and prevent DDoS attacks. The algorithm generates pre-processor rules in Snort for H-IDPS and uses a new detection and prevention mechanism to classify the NSL-KDD dataset, it discusses the evaluation of the classifier using a benchmark dataset. Finally, an evaluation process for the proposed mechanism is described in detail.

#### 3.2 Methodology Design Process

The main aim of this methodology design process is to achieve the three research objectives, as shown in Figure 3.1. Objective one is necessary to design a mechanism for H-IDPS. This mechanism will function as a classifier for protecting DDoS attacks. To do this, we examine the original ALO algorithm. Based on the work of (Mirjalili, 2015), the ALO benchmark outperforms other metaheuristic algorithms. Thus, we select ALO as the base algorithm, and attempt to overcome the weaknesses of MLP by feeding the weights through ALO ternaries. This will be explained in detail later in this section.

In Objective two is to implement the proposed ALO-MLP classifier, we will apply it to the NSL-KDD dataset through the topology of the cloud environment, which Snort is the source engine for H-IDPS. However, dimensional reduction removes redundant data, simplifying the process of sampling data for application to the classifier.

While, in objective three the proposed mechanism is evaluated using several metrics and the result are compared with those given by the main machine learning approaches from related work.

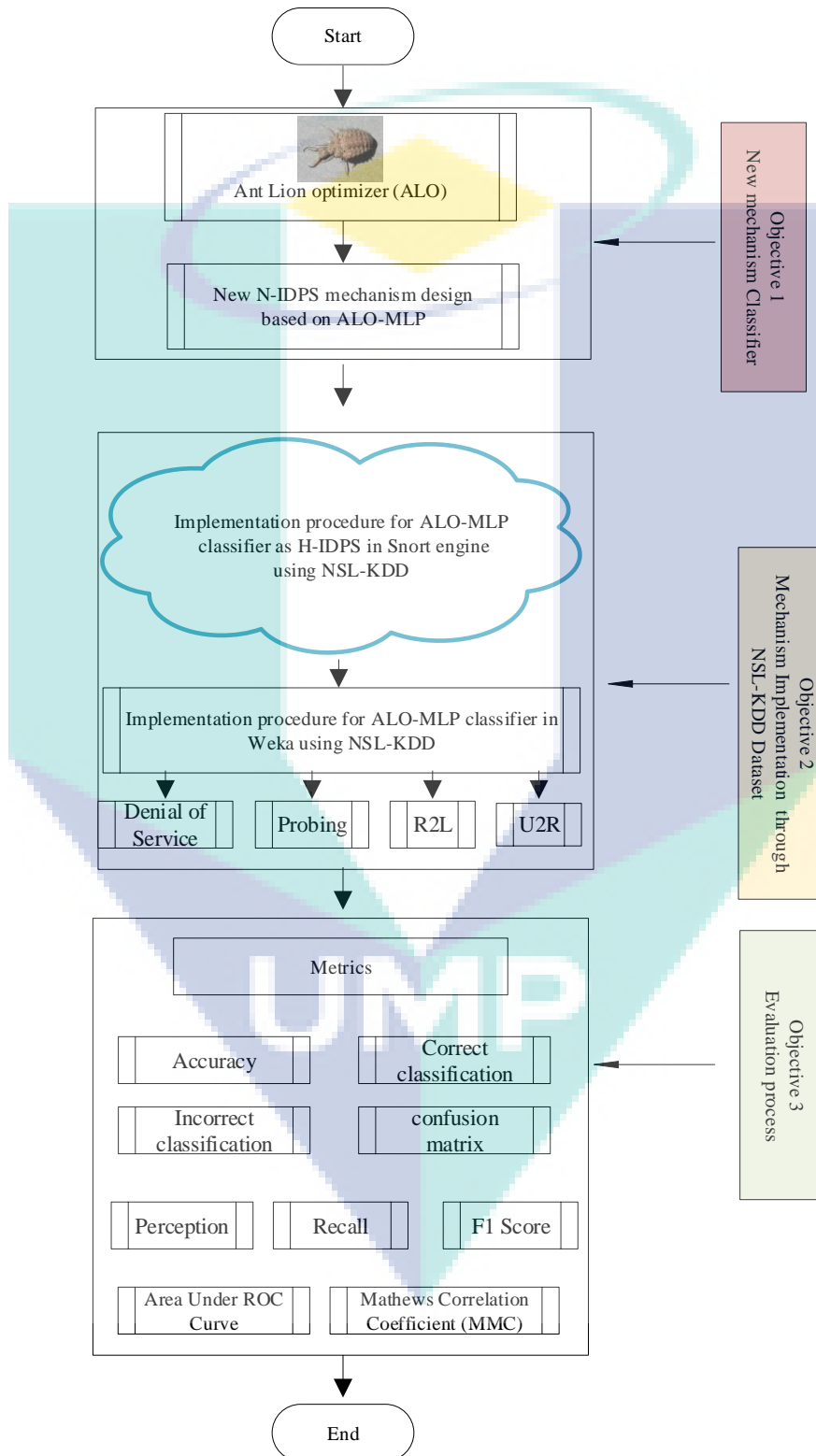


Figure 3.1 Proposed methodology and relation to research objectives.

### 3.2.1 Design of Classifier Mechanism

To achieve the objective1, a design a classifier has placed in Figure 3.2. Initially, we need to know the original algorithm for ALO before do the hybrid mechanism. Then, a detailed explanation for the ALO-MLP has done as shown in Figure 3.2.

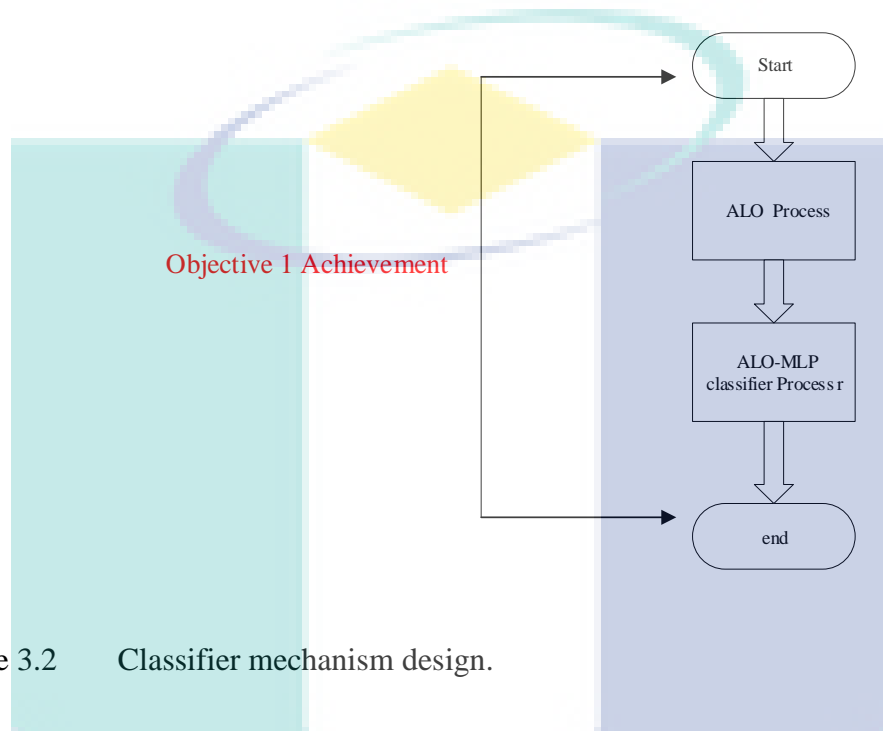


Figure 3.2 Classifier mechanism design.

#### 3.2.1.1 ALO Process

Antlions, sometimes known as doodlebugs, are part of the Myrmeleontidae family and go through larvae and adult life phases (Mani et al., 2018). As larvae, they have an interesting hunting mechanism whereby small cone-shape construction are used to trap ants. The antlions sit in a pit underneath the cones and wait for prey to be trapped, as shown in Figure 3.3 and the process flowchart in Figure 3.4. After consuming the prey's flesh, antlions throw the leftovers outside the pit and amend the pit for the next hunt. It has been observed that antlions tend to dig bigger pits when they are hungry, and this is the main inspiration for the ALO algorithm.

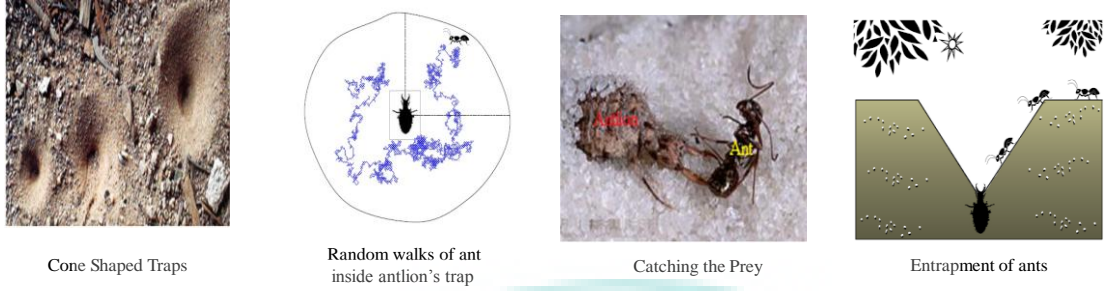


Figure 3.3 Operators of the ALO algorithm.

With the mechanisms proposed so far, antlions are able to build traps proportional to their fitness and ants are required to move randomly. However, antlions shoot sands outwards the centre of the pit once they realize that an ant is in the trap. This behaviour slides down the trapped ant that is trying to escape. For mathematically designing this behaviour, the radius of ant's random walks hyper-sphere is decreased adaptively. ALO is characterized as a three-tuple function, i.e.  $ALO(N_1, N_2, N_3)$ , that approximates the global optimum.  $N_1$ ,  $N_2$ , and  $N_3$  are formally defined as:

$$\eta \xrightarrow{N_1} \{\beta_{Ant}, \beta_{OA}, \beta_{Antlions}, \beta_{OAL}\} \quad 3.1$$

$$\{\beta_{Ant}, \beta_{Antlion}\} \xrightarrow{N_2} \{\beta_{Ant}, \beta_{Antlion}\} \quad 3.2$$

$$\{\beta_{Ant}, \beta_{Antlion}\} \xrightarrow{N_3} \{true, false\} \quad 3.3$$

where  $\beta_{Ant}$  is the ants' position matrix,  $\beta_{Antlion}$  contains the antlions' positions,  $\beta_{OA}$  includes the fitness of the ants, and  $\beta_{OAL}$  defines the fitness of the antlions. In this algorithm, the ant and antlion matrices are randomly initialized using the function  $N_1$ . The position of each ant with respect to the antlion is chosen by the roulette wheel operator, while the elite antlion position is updated by the function  $N_2$  in each iteration  $T$ . The boundary position is updated relative to the current iteration number. The position is then updated by two random walks around the elite and current antlion positions. The random path of each ant is evaluated by the fitness function. If any ant becomes fitter than an antlion, its position is considered as the new position for the antlion in the next iteration. The best current antlion is contrasted with the best antlion

obtained through optimization (elite). These steps are iterated until the function  $N_3$  returns a false value. The pseudocode for the ALO algorithm is as follows:

---

Algorithm 1: ALO

---

**Input:** Search-specific function, numbers of ants and antlions, number of iterations  $T$

**Output:** The elite antlion and its fitness

**1:** Initialize a population of  $n$  antlions and ants at random

**2:** Compute the ants' and antlions' fitness

**3:** Locate the best antlion and define as the elite

**4:** **While** the end condition is not satisfied

Foreach  $Ant_t$  do

Choose antlion utilizing roulette wheel selection

Slide ants toward the antlion

Create a random walk for the ant  
Update the position of  $Ant_t$  and normalize it,

End

**5:** Compute the fitness of all ants

**6:** Substitute an antlion for an ant if the ant is fitter

**7:** Update elite if an antlion becomes fitter than the current elite

**8:** **End While**

**9:** **Return** elite

---



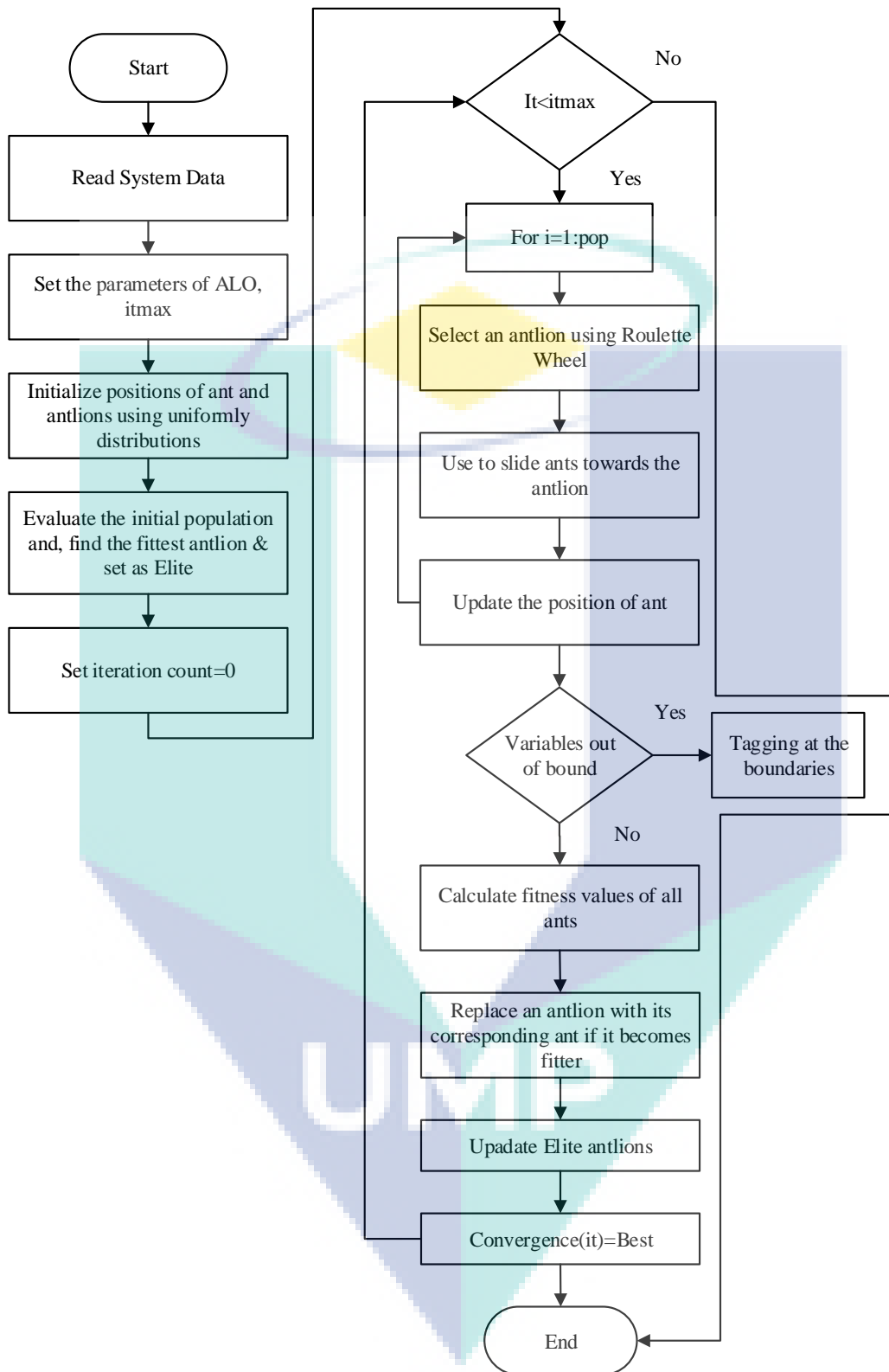


Figure 3.4 ALO process flowchart.

### 3.2.1.2 ALO-MLP Classifier Mechanism

The aim of any optimizer is to determine the variable values that give the highest classification rate and the lowest error rate. To achieve this, the ALO algorithm is used to optimize the weights and biases written in vector form, which represent the input to the ALO algorithm:

$$\vec{X} = \left\{ \begin{matrix} \vec{Y} \\ \vec{\theta} \end{matrix} \right\} = \{Y_{1,1}, Y_{1,2}, \dots, Y_{n,n}, \rho, \theta_1, \theta_2, \dots, \theta_h\} \quad 3.4$$

where  $n$  represents the number of inputs,  $Y_{ij}$  is the weight of the connection between the  $i^{th}$  and  $j^{th}$  nodes, and  $\theta_k$  represents the bias of the  $k^{th}$  hidden node. In other words, the objective of the proposed algorithm is to achieve the highest classification rate of both training and testing samples. To evaluate the MLP output, the Mean Square Error (MSE) is used to calculate the difference between the desired output and the actual output of MLP. The MSE is used to measure the deviation between the desired output and the actual output.

$$MSE = \sum_{i=1}^m (\kappa_i^k - \tau_i^k) \quad 3.5$$

where  $m$  represents the number of outputs, and  $\tau_i^k$ ,  $\kappa_i^k$  are the desired and actual outputs, respectively, of the  $i^{th}$  input unit when the  $k^{th}$  training sample is used. Thus, the average MSE can be calculated for all  $N$  training samples. The objective function of the ALO algorithm is used to minimize the average MSE as:

$$\min : F(\vec{X}) = \overline{MSE} \quad 3.6$$

Figure 3.5 illustrates how ALO feeds the weights and biases to produce more training sampling in an efficient manner.

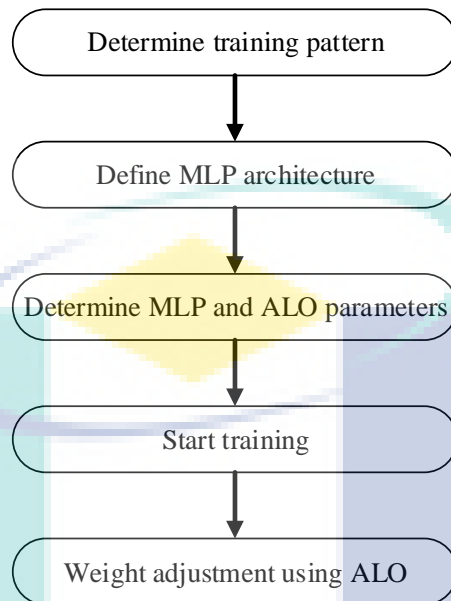


Figure 3.5 ALO-MLP mechanism.

Although most of the pre-processor rule options focus on simple checks against fields within Snort's packet structure, some of them are quite complex. In this section, these options are explored in more detail by examining how Snort evaluates the content rule option and its modifiers. The theory behind ALO-MLP and information on configuring and using the options will be highlighted. Building the pattern matcher begins with the pre-processor classifier that was previously used to evaluate the packets, as shown in Figure 3.6. The reason for using the pattern matcher is to reduce the number of rules that Snort must evaluate against the packet. Reducing the number of rules evaluated decreases the amount of time spent on any single packet. This allows Snort to process more packets and handle higher network speeds.

The pattern matcher starts by grouping rules based on their destination ports. For each pre-processing rule on a destination port, it then identifies the string with the longest content. If a rule does not have a content string, it is moved into a special non-content category. Once the strings have been collected, they are compiled into a set-wise pattern matcher using one of several possible algorithms. When a packet is input to the pattern matcher, the set of patterns for inspection is selected using the destination port. In a single pass, the pattern matcher then determines all patterns within the set that are contained within the packet. This pattern-matching process reduces the number of

rules considerably, thereby increasing the amount of traffic that Snort can analyse in semi-real time, as shown in Figure 3.6.

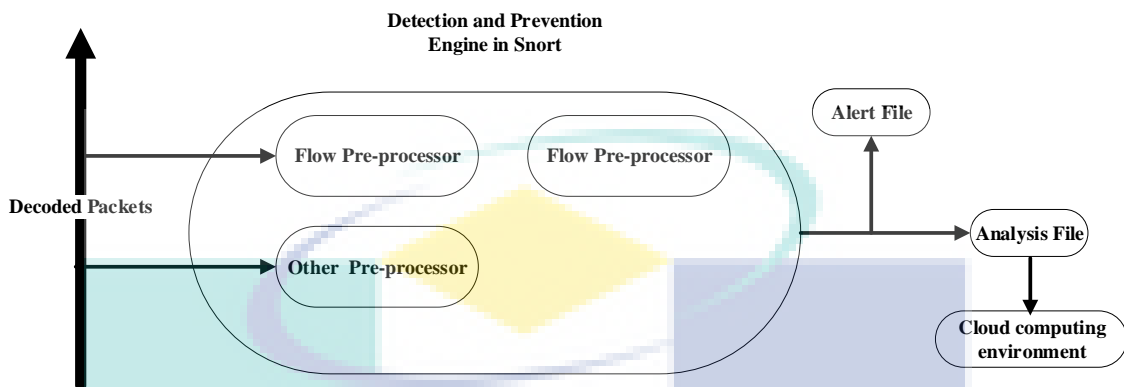


Figure 3.6 Detection and prevention engine in Snort.

As static profiles eventually become inaccurate, they are periodically regenerated. For example, an attacker may occasionally perform small amounts of malicious activity before subsequently increasing the frequency and quantity of this activity. If the rate of change is sufficiently low, the H-IDPS might treat the malicious activity as normal and include it in a normal profile. Malicious activity might also be observed by the H-IDPS while it is building its initial profiles.

Within the fast pattern matcher, the process of pattern matching does not consider positional modifiers such as the depth, offset, or distance that may be specified alongside the content option in the rule. These modifiers will be evaluated when the prevention engine calls the list of detection functions attached to the Option List (OTN) in the Snort engine. This improves performance, although using a long content match is not recommended if an efficient rule set is required. If the rule set grows beyond the memory available in H-IDPS, the options for matching become limited. However, work on an improved pattern matcher that offers similar performance as the current Adhocratic, which consumes only a small fraction of the memory, is currently underway.

Therefore, by using the fast pattern matcher, more complex configurations can be built, and the base Snort pre-processor rule set can be expanded. An initial profile is generated over a training period of several days or weeks. These profiles for the

prevention engine can be either static or dynamic. Once generated, a static profile is constant unless the H-IDPS is specifically directed to generate a new profile.

Dynamic profiles are constantly adjusted as additional events are observed. The corresponding measure of normal behaviour could be different, as the systems and networks will change over time as shown in Figure 3.7

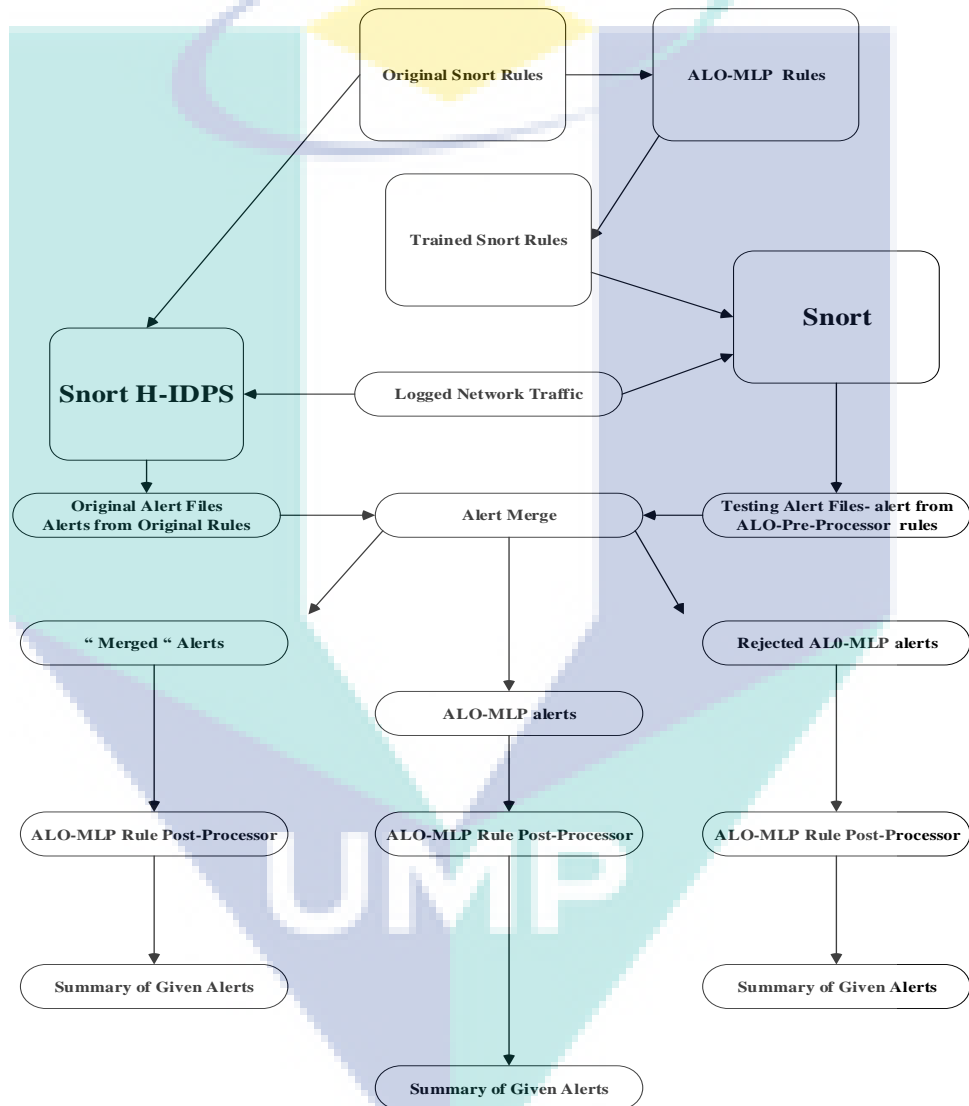


Figure 3.7 ALO-MLP as a pre-processor classifier in Snort core.

After being introduced to the cloud environments hypervisor based on any suspicious IPs, the H-IDPS will take immediate action against anomalous DDoS packets. The hypervisor will create a list of discrete entities such as the hosts, TCP, and UDP port numbers associated with the malicious activity, which extracted from NSL-KDD traffic in Snort. Blacklists, also known as hot lists, are typically used to allow the

H-IDPS to recognize and block activity that is highly likely to be malicious; this list may also be used to assign a higher priority to alerts that match entries on the blacklists.

The proposed mechanism generates dynamic blacklists that are used to temporarily block recently detected threats (e.g. activity from the IP address of an attacker). A whitelist is a list of discrete entities that are known to be benign. Whitelists are typically used on a granular basis, such as to reduce or ignore false negative involving known benign activity from trusted hosts in true positive. Whitelists and blacklists are commonly used in signature-based detection and stateful protocol analysis.

### 3.2.2 Implementing ALO-MLP as a Classifier for the NSL-KDD Dataset

Packets captured from different source of NSL-KDDIP addresses are recognized as attack packets, whereas the aggregation of packets sent from different sources to a specific destination is classified as a DDoS attack, which extracted from NSL-KDD. The incoming traffic from different sources to one specific destination sends the result to the correlation engine to detect the DDoS attack.

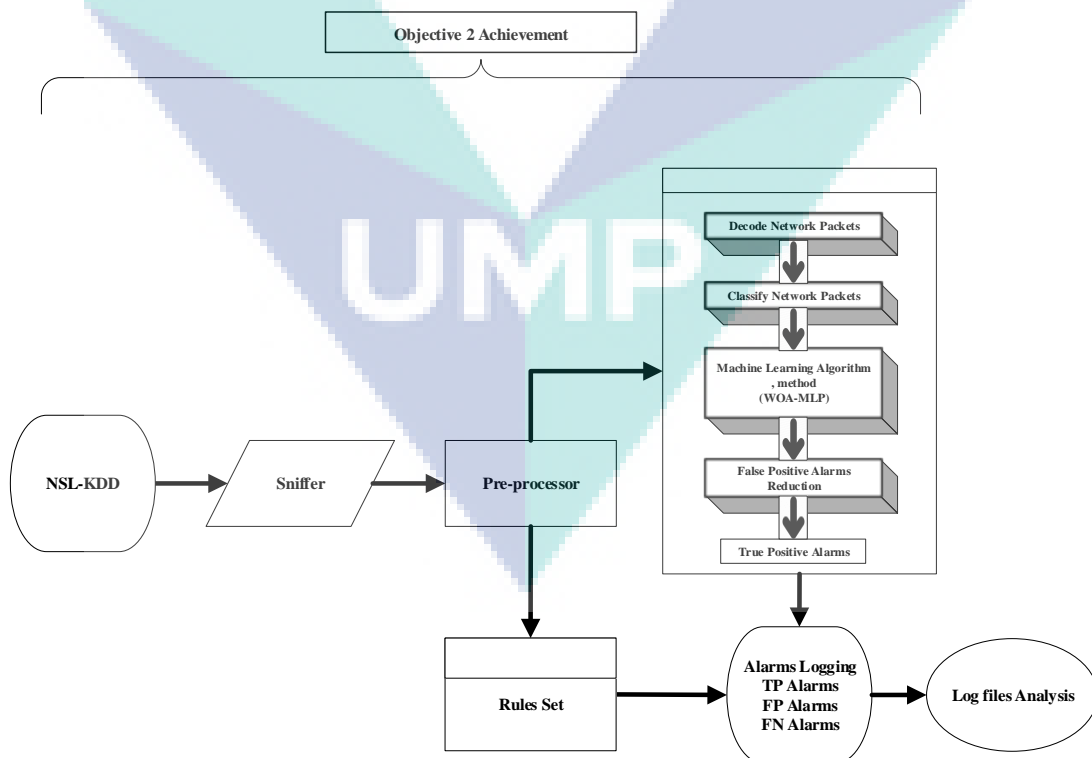


Figure 3.8 Implementing ALO-MLP.

There are two focal issues that need to be addressed for an H-IDPS: cleaning the training data and devising an enriched representation for the mechanism. Strategies for both could improve the performance of an anomaly detection system. Techniques that perform system monitoring require clean training data when building their mechanism. The current audit sequence is then examined for anomalous behaviour using some supervised learning algorithm.

An attack embedded inside the training data would result in an erroneous mechanism, as all future occurrences of that attack would be treated as normal. Moreover, obtaining clean data by hand could be tedious. Hence, an automated technique for purging all malicious content from audit data is required. Additionally, normal behaviour is structured using features extracted from the training set. It is important to remember that the concept of normal/abnormal in anomaly detection is vague when compared to a virus detector, which has an exact signature of the virus it is trying to detect. As a result, anomaly detection is a difficult problem.

Traditional host-based anomaly detection systems focus on system call sequences to build mechanisms of normal application behaviour. These techniques are based upon the observation that malicious activity results in an abnormal (novel) sequence of system calls. Several variants of LDA have been used to address the vanishing of within-class scatter under the projection to a low-dimensional subspace. However, some of these proposals are ad-hoc and do not address the generalization problem for new data (Rathore et al., 2016).

Although LDA is preferred in several dimension reduction applications, it does not always outperform PCA. Therefore, both PCA and LDA are combined to optimize the performance of NSL-KDD. The main goal is to enhance data discrimination, which can be achieved with subspaces learned with either PCA or LDA, as shown in Figure 3.9.

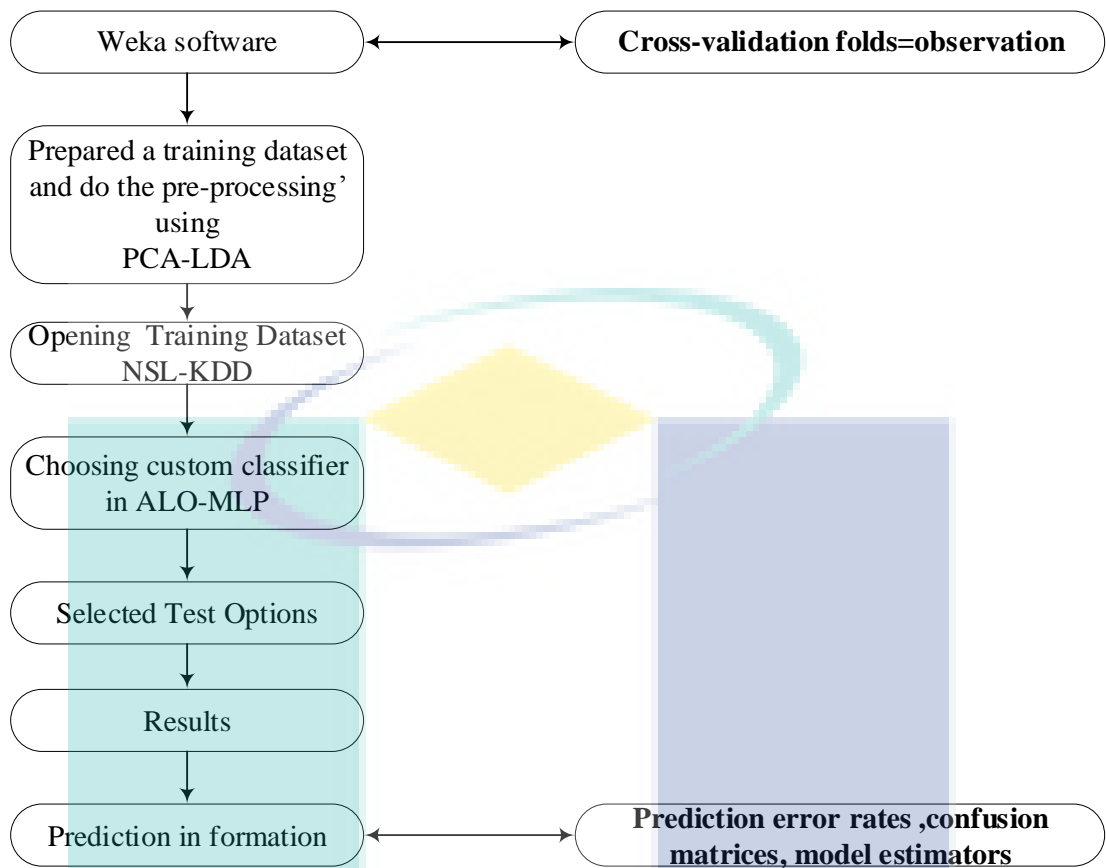


Figure 3.9 ALO-MLP classifier testing in Weka.

The learning mechanism of the hybrid method differs from those of existing techniques. The hybrid mechanism addresses the generalization problem for new data directly, a novel computational strategy has been developed to estimate the optimal subspaces. Given a set of labelled training data from different classes and a set of unlabelled test data from the same group of classes, each test sample is identified using the new mechanism. Both sets consist of feature vectors. The hybrid procedure for PCA and LDA is described below:

1. Compute Mean vectors for the input features dataset ( $x_i$ )

$$\text{Mean } \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad 3.7$$

2. Calculate the scatter matrix – Covariance Matrix

$$S = \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T \quad 3.8$$

3. Compute Mean vectors for the principal components ( $Px_i$ )

$$\text{Mean } \bar{Px} = \frac{1}{n} \sum_{i=1}^n Px_i \quad 3.9$$



4. Calculate the scatter matrices – within class( $S_w$ ) and between class( $S_B$ ) matrices

$$S_w = \sum_{i=1}^n (P x_i - \bar{P x})(P x_i - \bar{P x})^T \quad 3.10$$

$$S_B = \sum_{i=1}^n N_i (\bar{P x} - m)(P \bar{x} - m)^T \quad 3.11$$

5. Find linear discriminants by computing the eigen values for  $S_w^{-1} S_B$
6. Select the linear discriminants for the new feature set by sorting and choosing eigen vectors,  $W$  with highest eigen values.
7. The new feature set obtained by the linear discriminants are then used to obtain transformed input dataset by following equation

$$Y = X.W \quad 3.12$$

### 3.2.2.1 Scenario 1: Denial of Service

This attack scenario is designed to perform attacks on a target using the targa8 tool until it is successful. Targa is a very powerful tool that can quickly damage a network belonging to an organization.

### 3.2.2.2 Scenario 2: Probing

In this scenario, we attempt to acquire information about the target host and then launch an attack by exploiting vulnerabilities found using the nmap9 tool. Examples of attacks that can be launched by this method are SYN-scan and ping-sweep.

### 3.2.2.3 Scenario 3: R2L

This scenario has the goal of performing coordinated port scans to single and multiple targets. The tasks are distributed.

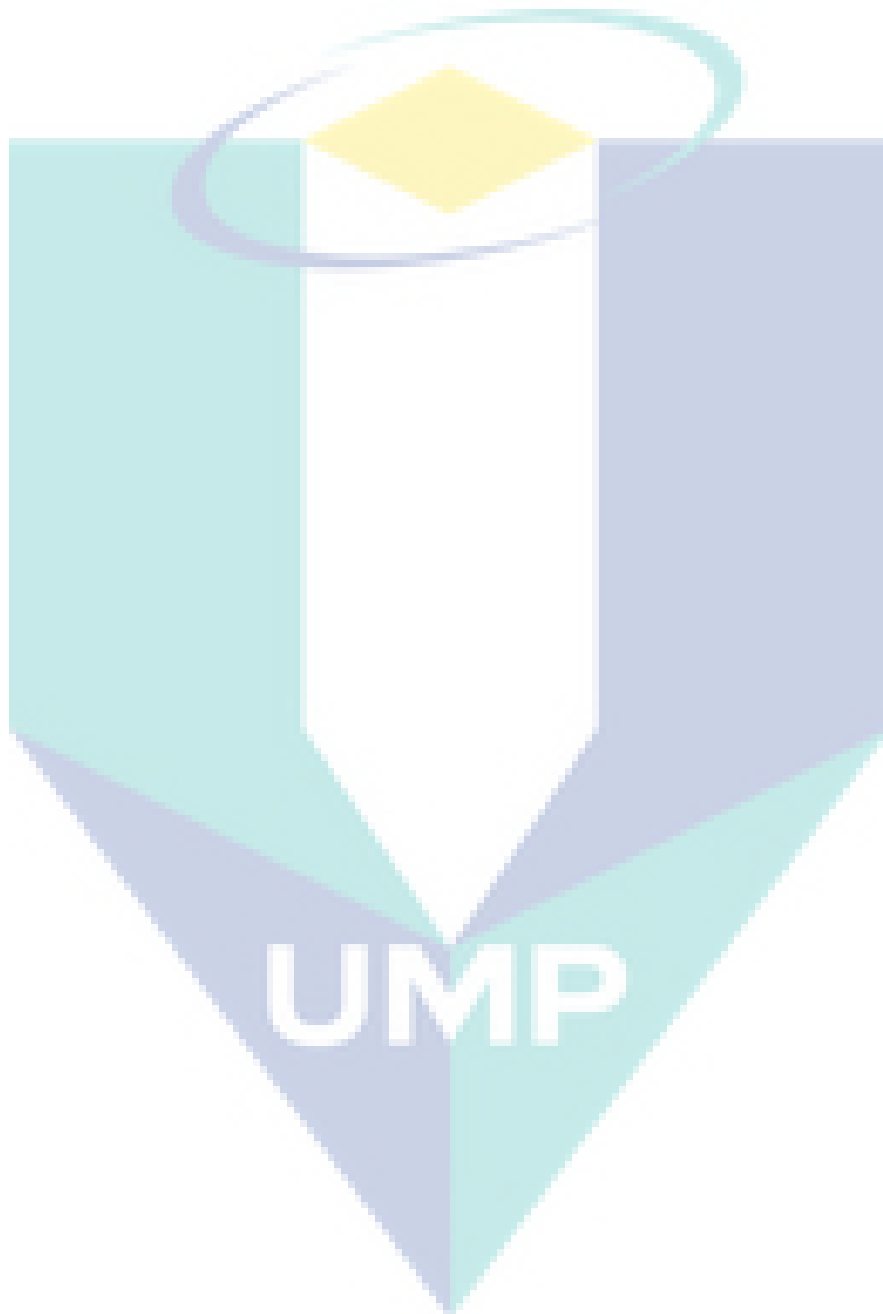
### 3.2.2.4 Scenario 4: User to Root

These attacks are very common against networks, as they tend to break into accounts with weak username and password combinations.

## 3.2.3 Performance evaluation of the Proposed Mechanism

A binary classifier produces output with two class values or labels, such as Yes/No and 1/0, for given input data. The class of interest is usually denoted as “positive” and the other as “negative”. NSL-KDD used for performance evaluation is called a test dataset. It should contain the correct labels (observed labels) for all data

instances. These observed labels are used to compare with the predicted labels for performance evaluation after classification. Figure 3.10 shows the nine-evaluation metrics that used to evaluate the proposed mechanism.



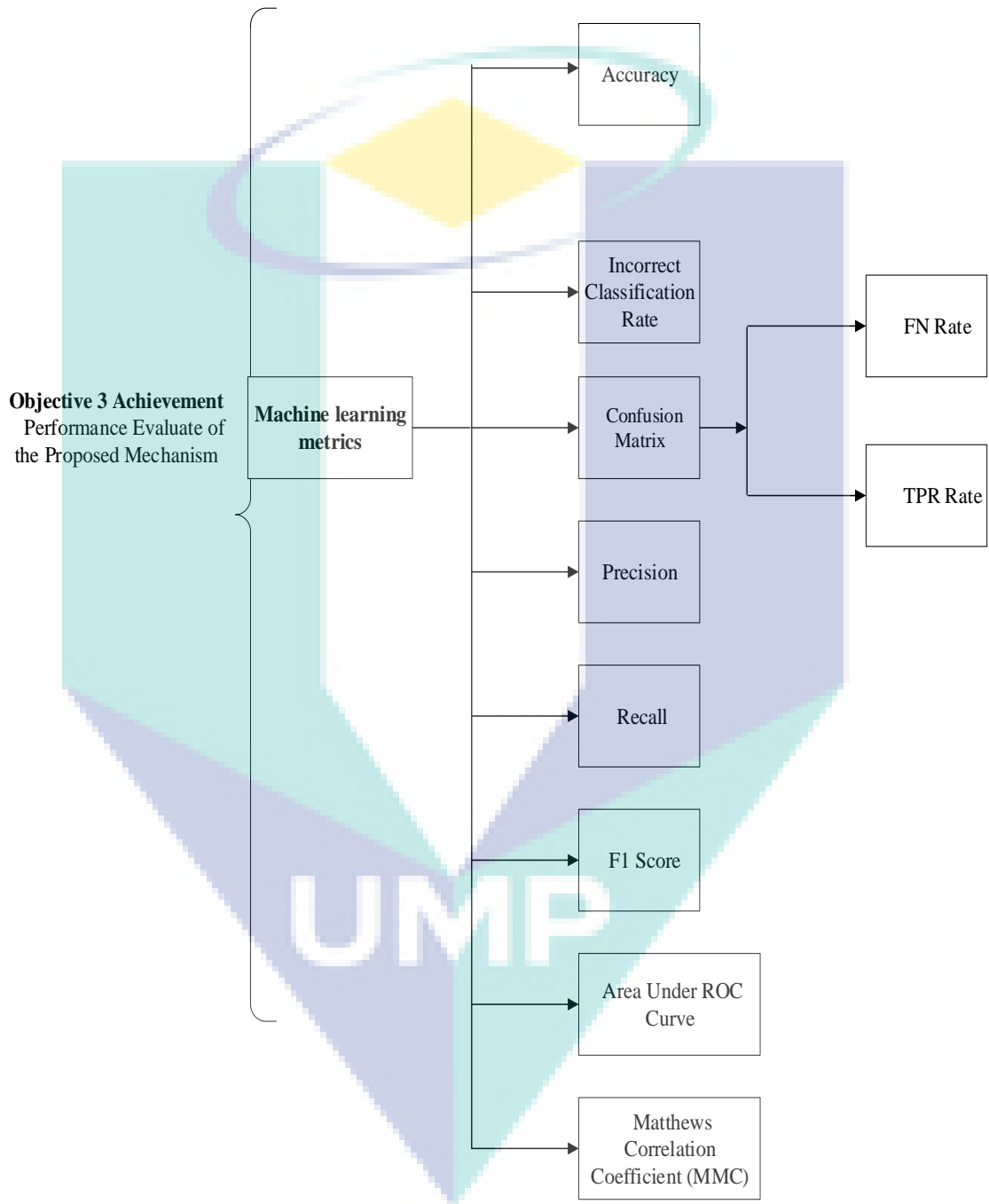


Figure 3.10 Evaluation Metrics

### 3.2.3.1 Accuracy

The accuracy (*ACC*) is the proportion of correct predictions over the whole dataset. It is determined as:

$$ACC = (TP + TN) / (TP + TN + FP + FN) \quad 3.13$$

### 3.2.3.2 Incorrect Classification Rate

Misclassification is the situation where an intrusion is assigned a class (either normal or anomalous) that is different from the actual one (Salman et al., 2017). This measure is useful in estimating the probability of disagreement between the true and predicted classification rates of H-IDPS. It is obtained by dividing the sum (FN+FP) by the total number of paired observations, i.e. TP+FP+FN+TN.

In a binary classifier, we only have the two classes of “Attack” and “Normal”. Thus, we have four instances: an “Attack” predicted as “Attack” (TP) or predicted as “Normal” (FN), and “Normal” predicted as “Normal” (TN) or predicted as “Attack” (FP). The classifiers used with the NSL-KDD dataset aim to predict the class of each attack. If an attack that belongs to a certain class is incorrectly predicted as belonging to the wrong class of attacks, it has still been correctly identified as an attack. Thus, we cannot consider this case as a false positive or a false negative, as it contradicts the definition of both.

### 3.2.3.3 Confusion Matrix

A confusion matrix can be used to illustrate the performance of an H-IDPS. The confusion matrix can be used for N-class problems, whereas the matrix discussed earlier is used for 2-class problems (Chatterjee & Bhattacharya, 2014). The size of the matrix depends on the number of distinct classes to be detected in the dataset. The matrix entries reflect a comparison of the class labels predicted by the classifier and the actual class labels. Consider an intrusion dataset with 100 instances, of which 45 are normal, 35 are DDoS, 15 are probe, and 5 are U2R attack instances. Of the 45 normal instances, the H-IDPS predicts 38 correctly, 4 as U2R, 2 as probe, and 1 as DDoS; of the 35 DDoS instances, the N-IDS predicts 31 correctly and 4 as normal; of the 15 probe instances, the system predicts 11 correctly and 4 as normal; and finally, of the 5 U2R instances,

the system predicts 3 correctly, 1 as normal, and 1 as probe. The confusion matrix for this situation is illustrated in Figure 3.11.

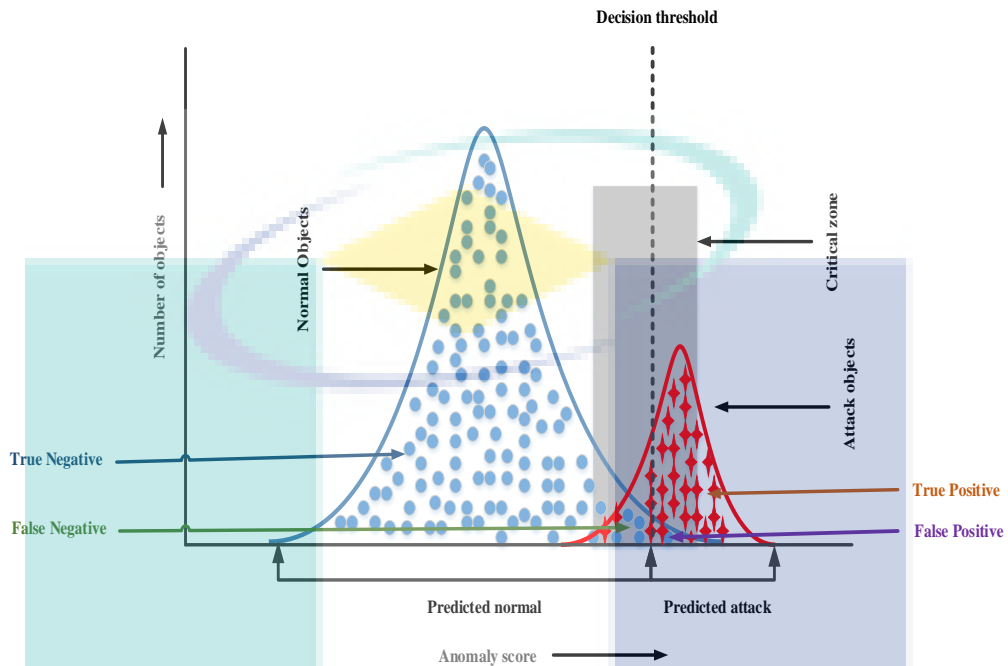


Figure 3.11 Confusion matrix classification.

The Receiver Operating Characteristic (ROC) curve could not be used to compare the IDPS and determine which is most suitable for certain circumstances. For example, the appropriateness of ROC analysis is very questionable when the IDPS only produces 0 or 1 as output, and the proper unit of analysis and measurement differs for different detectors.

### 3.2.3.4 Precision

The precision (P) is the proportion of attack cases that are correctly predicted relative to the predicted size of the attack class. This is calculated as:

$$P = \frac{TP}{(TP+FP)} \quad 3.14$$

### 3.2.3.5 FN Rate

In statistics, when performing multiple comparisons, a false negative ratio (or false alarm ratio) is the probability of falsely rejecting the null hypothesis for a test. The false negative rate is calculated as the ratio between the number of negative events

wrongly categorized as positive (false positives) and the total number of actual negative events (regardless of classification).

The false negative rate (FN) is calculated as:

$$FN = FN / (TP + FP) \quad 3.15$$

### 3.2.3.6 Recall

Recall (also known as sensitivity) is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. Both precision and recall are therefore based on an understanding and measure of relevance. The recall (R) or TP rate (TPR) is the proportion of correctly predicted attack cases over the actual size of the attack class. This is calculated as:

$$\text{Recall} = FN / (TP + FN) \quad 3.16$$

### 3.2.3.7 F1 Score

In the statistical analysis of binary classification, the F1 score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision P and recall R of the test to compute the score: P is the number of correct positive results divided by the number of all positive results returned by the classifier, and R is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). The F1 score is the harmonic average of the precision and recall. The best F1 score is 1 (perfect precision and recall) and the worst is 0.

$$F1 = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall}) \quad 3.17$$

### 3.2.3.8 TPR

The attack or anomalous data are labelled as positive, whereas normal data are labelled as negative. The true label is used for decision-making. We have the four possibilities of TP, TN, FP, and FN (Li et al., 2014). When the H-IDPS correctly

classifies an anomalous instance, this counts as a TP. FP occurs when a legitimate action is misclassified as anomalous.

TPR = true positive fraction

= 1 – false negative fraction

= TP / (TP + FN)

3.18

### 3.2.3.9 Area Under ROC Curve (AUC)

Figure 3.12 shows three AUC curves representing excellent, good, and worthless tests. The accuracy of the test depends on how well the test separates the group being tested into those with and without the attribute in question. Accuracy is measured by the area under the ROC curve. An area of 1 represents a perfect test; an area of 0.5 represents a worthless test.

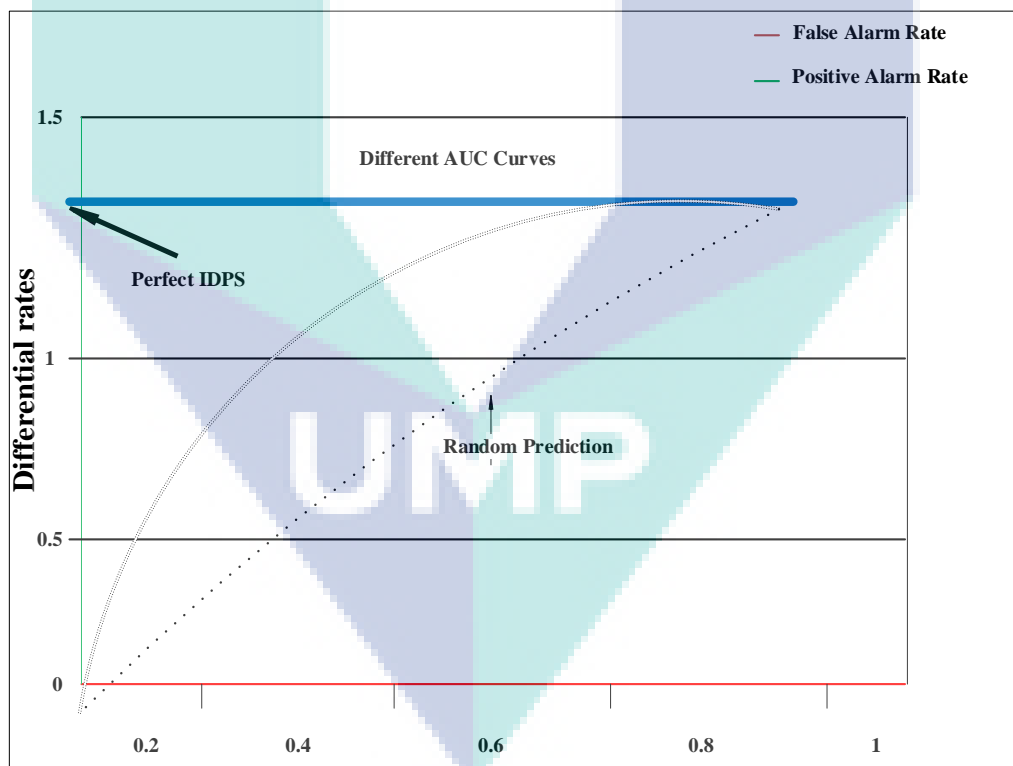


Figure 3.12 Illustration of the area under the AUC curve.

### 3.2.3.10 Matthews Correlation Coefficient

The Matthews correlation coefficient (MCC) is used in machine learning to measure the quality of binary (two-class) classifications (Boughorbel et al., 2017). However, it considers true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC describes the correlation between the observed and predicted binary classifications, returning a value between  $-1$  and  $+1$ . A coefficient of  $+1$  represents a perfect prediction,  $0$  is no better than random, and  $-1$  indicates total disagreement between prediction and observation. The MCC is calculated as:

$$MMC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad 3.19$$

### 3.3 Chapter Summary

This Chapter has described the mechanism based on two important methods: a new Snort classifier based on ALO-MLP for H-IDPS in cloud environment, and the ALO-MLP classifier in Weka. The proposed use of ALO-MLP in H-IDPS is based on new mechanism design with the hypervisor environment to detect and prevent DDoS attacks. Therefore, achieving the objectives stated in Chapter 1. The next Chapter describes the implementation of the proposed solution for the dataset.



## CHAPTER 4

### IMPLEMENTATION AND RESULTS

#### 4.1 Overview

This Chapter describes all implementation processes of the proposed prevention mechanism for DDoS attacks in a cloud environment through the H-IDPS inline hypervisor. Performance results using the proposed implementation are also presented for various scenarios. Section 4.2 introduces the implementation of NSL-KDD and extracted dataset into Snort over cloud environment and Weka. The parameters and metrics used in the implementation stage are reported in Section 4.3. Section 4.4 presents the NSL-KDD results though Weka. Furthermore, a method for identifying which IPs are attacking the cloud server is described in section 4.5. Finally, Section 4.6 summarizes the current Chapter.

#### 4.2 Implementation Phases

The Snort pre-processor rule sensor operates efficiently in cloud environment. The mechanism was used over NSL-KDD. However, the selected classifier was tested using 74,637 records, which were fully randomized to simulate a more realistic situation. All network traffic was collected and audited by H-IDPS as follows:

1. Classifier mechanism for ALO-MLP implemented in Weka.
2. ALO-MLP classifier verification using NSL-KDD.
3. Implementation of evaluation metrics for ALO-MLP.

### 4.2.1 Implementation of Designed ALO-MLP Classifier

H-IDPS was built as a core network security system for the cloud computing environment. Thus, H-IDPS was used to detect and prevent all types of DDoS attack in NSL-KDD. This allows the hypervisor to profile the results in white/black lists. This implementation was intended to create and test the virtual network in a small-scale environment. A simple virtual network was created and tested with the help of a single virtual router, virtual switches, and two virtual computers as hosts in VMware Workstation 14. The second part of this stage focused on implementing and scaling the virtual network to large numbers of devices using ESXi. In this part, a large-scale virtual network was created with multiple switches and a router, and then installed in the server environment. The router and switches erased the previous configuration automatically.

There are specific instructions regarding how new pre-processor plugin modules can be incorporated. The ALO-MLP generic flood detection and prevention pre-processor was named ALO-MPL\_flood.c and was accompanied by the ALO-MLP\_flood.h header file, where employed as DDoS prevention. The following statements describe the initial steps required to add the ALO-MLP\_flood pre-processor to Snort:

Add to the Snort plugbase.h file

```
#include "ALO-MLP_flood.h"
```

Add the following lines to the Snort plugbase.c file

```
void Init. Pre-processor ()
{
    SetupFlood();
}
```

Add the following lines to the Snort. conf file

```
pre-processor flood: $HOME_NET <threshold # packets> <threshold # time period>
<logfilename>
```

Create two flood-plugin files:

ALO-MLP\_flood.h

ALO-MLP\_flood.c

In ALO-MLP\_flood.h, add

```

void SetupFlood ();
void FloodInit (u_char *);
# The FloodInit function creates the pre-processor data structure
In ALO-MLP_flood.c, register the pre-processors by adding the following function:
void SetupFlood(void)
{

```

```

    Register Preprocessor ("flood", FloodInit);

```

#### 4.2.1.1 Flood Pre-Processor Data Structure

The pre-processor flood list maintains the packet rate using a three-dimensional double-linked list:

1. Flood list → source info (match source IP).
2. Destination info → (match destination IP).
3. Connection info → (match port info).

The first-level source info list registers the packet source address. For each source, the packet's destination was recorded and counted in the destination info list. For each source–destination connection, the packet's port information was recorded and incremented. The key data structures used for flood detection are presented in Figures 4.1 and 4.2.

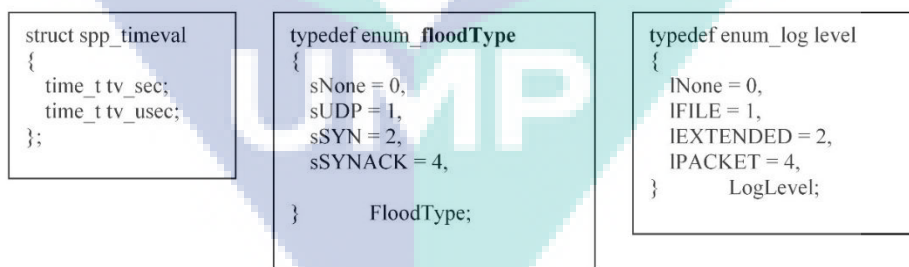


Figure 4.1 DDoS pre-processor key data structure.

Based on the ALO-MLP pre-recorded rules, H-IDPS classified and prioritized Snort alerts. This process is fully customizable and allows the desired classifications and priorities to be defined. There are three priority levels by default: low, medium, and high.

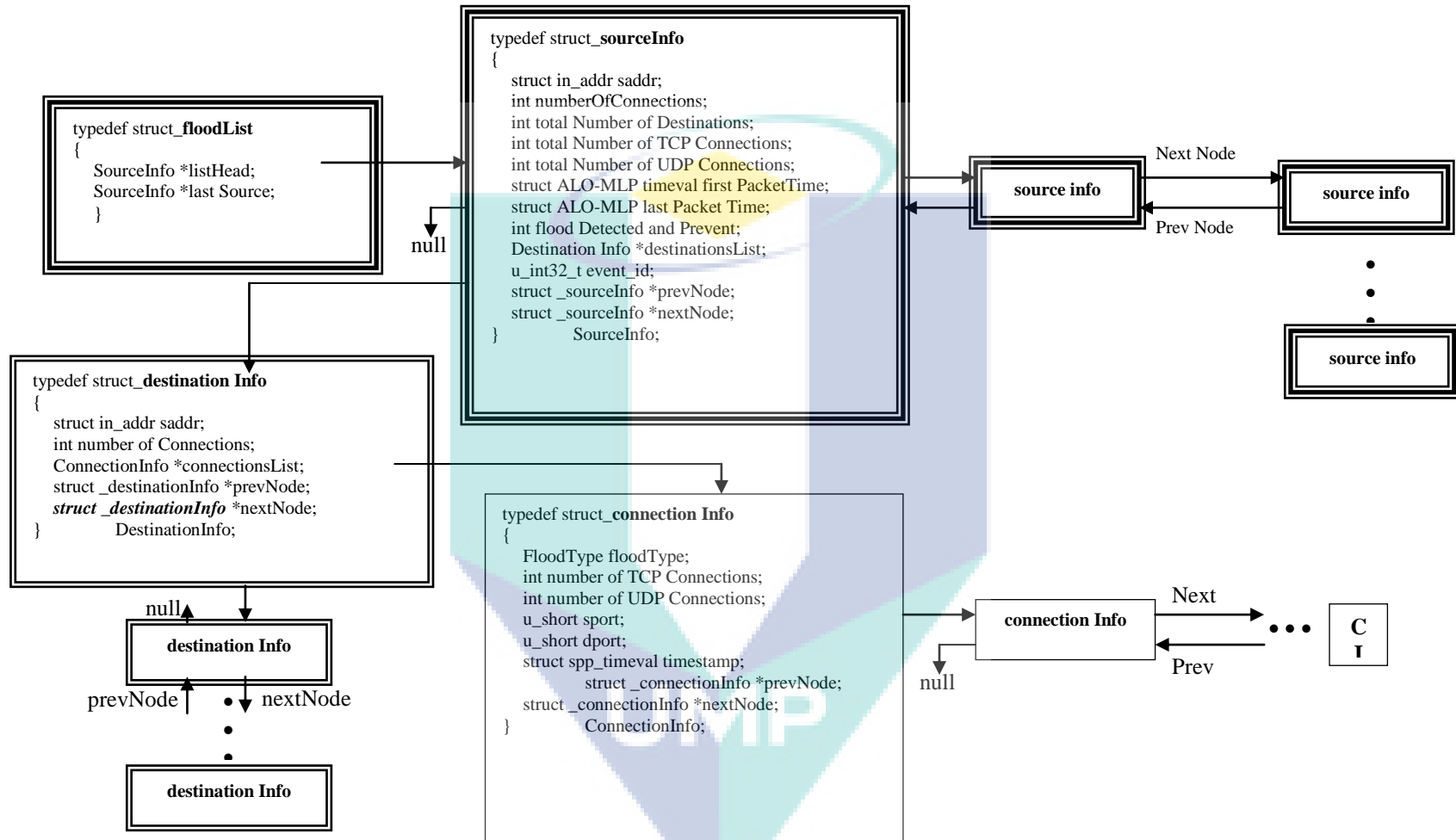


Figure 4.2 DDoS pre-processor key data structure.

The computer used for attack coordination assumed the dual roles of “Master client” and “Handler”. The system specifications and those of four attack agents are presented in Table 4.1; the H-IDPS cloud hypervisor testbed is shown in Figure 4.3.

Table 4.1 System specifications

System	Specifications
Model	VL HP ProLiant dl320e gen8
CPU	Intel® Xeon® E3-1220Lv2 (2.3GHz/2-core/3MB/17W, HT)
RAM	16 Gigabyte 1600MHz DDR3
Hard Drive	2 Terabyte
Network Interface	HP Ethernet 1Gb 2-port 330i Adapter
OS	Debian 9.2
Linux Kernel Version	4.9 2017



Figure 4.3 Two HP servers.

#### 4.2.1.2 Cloud Environment H-IDPS Network System Specifications

The hypervisor H-IDPS used identical computer models as described in Table 4.1 with Snort v2.9.11. However, the firewall overrides the security policy and routes TCP and UDP external traffic-which came from NSL-KDD to specific service ports on the testbed, namely TCP/UDP-ports 21, 22, and 23 for FTP, SSH, and Telnet services.

- TCP/UDP-port 25 for SMTP services.
- TCP/UDP-port 42 for DNS services.
- Other ports opened for remote administration of Real Server, Snort, and other applications.

For testing purpose, iptables were also configured to allow ICMP packets. Class-based queuing was implemented on Titan to manage outbound traffic into the private

network. Seventy percent of the internal link bandwidth was allocated to HTTP and RealPlayer traffic. Moreover, SMTP was assigned 15% of the bandwidth, and SSH, Telnet, and FTP collectively used 10% of the bandwidth. Finally, SYN and ICMP traffic was bounded to 5% of the network link bandwidth. The rateif.pl program in Titan opens port 6779 to listen for alerts from H-IPDS.

#### 4.2.2 Implementation of ALO-MLP H-IDPS Classifier using NSL-KDD

We used nine algorithms in Weka for the classification task. The test option used in all techniques was 10-fold cross-validation. NSL-KDD is suggested to solve some of the problems in the original KDD99 dataset. One of the most important deficiencies in the KDD data set is the huge number of redundant records, which causes the learning algorithms to be biased towards the frequent records. Furthermore, the number of records in the NSL-KDD train and test sets are reasonable. This advantage makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, Data files in NSL-KDD have:

1. **KDDTrain+.ARFF:** The full NSL-KDD train set with binary labels in ARFF format.
2. **KDDTest+.ARFF:** The full NSL-KDD test set with binary labels in ARFF format.

However, the normal traffic in this dataset was 972,781 records and attacks in was 3,925,650. Whereas, the dataset contains 41 features which are listed in the Table 4.2.

In each record there are 41 attributes unfolding different features of the flow and a label assigned to each either as an attack type or as normal. The details of the attributes namely the attribute name, it contains type information of all the 41 attributes available in the NSL-KDD data set as shown in Tables 4.2. In addition, these attributes contain data about the various 5 classes of network connection vectors and they are categorized as one normal class and four attack class. Tables 4.3, shows the 4 attack classes are further grouped as DDoS, Probe, R2L and U2R. The description of the attack classes.

Table 4.2 NSL-KDD features.

S.NO	Class	Features
1. Basic features	F1-F9	Duration Protocol Type, Service Flag, Source Bytes, Destination Bytes, Land Wrong Fragment, Urgent.
2. Content features	F10-F22	Number Failed Logged In, Number Compromised, Root Shell, Su Attempted, Number Root, Number File Creations, Number Shells, Number Access Files, Number Outbound Cmds, Is Host Login
3. Same host features	F23-F31	Count, Srv Count, Serror Rate, Rerror Rate, Srverror Rate, Same Srv Rate, Diff Srv Rate, Srv Diff Host Rate.
4. Same services features	F32-F41	Dst Host Count, Dst Host Srv Count, Dst Host Same Srv Rate, Dst Host Diff Srv Rate, Same Scr Port Rate, Dst Host Srv Diff Host Rate, Dst Host Serror Rate, Dst Host Srverror Rate, Dst Host Rerro Rate, Dst Host Srverror Rate

Table 4.3 Attack type and their related attack.

Category	Attacks
<b>DDoS</b>	Duration, Scr-bytes, Count,, Srv_rerror_rate , ,dst_host_same_srv_rate ,dst_host_srv_serror_rate, ,dst_host_srv_rerror_rate
<b>Probe</b>	http tunnel, ftp_write, multihop, buffer overflow, root kit, xterm, ps.
<b>R2L</b>	guess_passwd, named, snmpgetattack, xlock, send mail
<b>U2R</b>	ipsweep, nmap, port sweep, satan, mscan, saint

### 4.3 Evaluation of Mechanism

Forty-one features were calculated and stored in the database. The attack features included the number of DoS, Probe, U2R and R2L. The number of features was then reduced using PCA (Badis et al., 2014). It is important to characterize and evaluate the reliability (precision) of the solution under different conditions. Hence, the performance evaluation of machine learning systems is increasingly popular. H-IDPS can be evaluated using AUC, which depicts the trade-off between the TP and the FN rate.

#### 4.3.1 Parameters

In the case of the proposed mechanism, the network packets were decoded, their header fields were evaluated, and the relevant features were computed with respect to some statistical properties of the traffic. Table 4.4 lists the parameters used in the H-IDPS Weka setup.

Table 4.4 Control parameters used in H-IDPS.

Parameter type	Value
Multimodal Benchmark Functions	15-Dimensional
Trainfcn	Trainscg
Hidden Layer Size	10
Training Ratio	70%
Validation Ratio	15%
Test Ratio	15%
Simulation Time	Five Minutes for Single Run
Simulation Run	100 Times
DDoS Report Output Format	Excel Sheet
Number of Search Agents	3000
Population (No. Of Ants) (N)	50
Maximum Iteration Count (T)	500
Number of Variables (DIM)	6
Random Number	[0, 1]



### 4.3.2 Confusion Matrix

The correctness of a classification can be evaluated by computing the number of correctly recognized class examples (TP), the number of correctly recognized examples that do not belong to the class (TN), and examples that are either incorrectly assigned to the class (FP) or not recognized as class examples (FN).

### 4.4 ALO-MLP classifier Scenario Results Through NSL-KDD

The results from the proposed H-IDPS under NSL-KDD classes attack are now presented. Several machine learning metrics and parameters have been adopted. Finally, the parameters that produced the highest detection rate are reported. For brevity, we focus on the case of sample confusion matrices as shown in Table 4.5.

Table 4.5 Sample confusion matrix for ALO-MLP for 74,637 samples.

	Positive	Negative
Positive	4594	13
Negative	17	3088

#### 4.4.1 DoS Scenario Results

For DoS attacks, the features ‘land’ and ‘urgent’ were removed due to their lack of information gain. After ALO-MLP classified it using Weka. The remaining features are 1–18, which shows that there are still strong correlations between some higher-level features. Several different well-performing subsets were extracted. Using training set with 10-fold cross-validation, shows that the removal of any feature leads to performance loss.

Based on our performance metrics, Table 4.6 shows that it clearly indicates that ALO-MLP classifier could be the most suitable mechanism to reduce the FN and increase the accuracy. Additionally, ALO -MLP has also achieved the first rank in DoS class of attack in terms of accuracy, incorrect classification, TP, FN, precision, Recall and F-measure. Regarding accuracy, it comes out (99.1% correct predictions out of 100 total examples). That means our classifier is doing a great result of identifying DoS attack.

FN result shown as (0.005%), which is the lower rate. However, TP rate was 97.9%, which considerably good for our mechanisms. ALO-MLP has highest precision rate 97.65%, which means that it has the FN value. Furthermore, the recall percentage 97.9%, which means that it has the lowest FN (missed attacks) percentage.

It is important to note that MMC usually gives lower values which is only percentage (1%). Using the ALO-MLP to classify new incoming traffic is very fast, which is the main benefit of this method. However, best performance AUC for testing with percentage (1%).

Table 4.6 ALO-MLP classifier for DoS over metrics.

<b>Evolution</b>	<b>DoS (%)</b>
Accuracy	99.1
Incorrectly Classified Instances	0.389
TP Rate	99.7
FN Rate	0.005
Precision	99.6
Recall	99.7
F-Measure	99.2
MCC	1.000
AUC	1.000

#### 4.4.2 U2R Scenario Results

Features with information gain are 1–5 and ‘num file creations’ is the most important feature. Table 4.6 shows that it clearly indicates that ALO-MLP classifier in U2R scenario has reduced the false alarm and increase the accuracy. Additionally, ALO-MLP has also achieved U2R class of attack in terms of accuracy, incorrect classification, TP, FN, precision, Recall and F-measure. Regarding accuracy, it comes out (97.37%), which means our classifier is efficient enough to identifying U2R attack.

FN results increase in overheads, may cost time and resources of systems but it shown as (0.030%), which is the lower rate. TP rate percentage (97.9%), which considerably good for our mechanism. ALO-MLP has highest precision percentage (96.2%), Furthermore, the recall percentage (97.9%).

It is important to note that MMC usually gives lower values which is only percentage (94.7%). Using the ALO-MLP to classify new incoming traffic is very fast,

which is the main benefit of this method. However, better performance AUC for testing with percentage (99.8%).

Table 4.7 ALO-MLP classifier for U2R over metrics

<b>Evolution</b>	<b>U2R (%)</b>
Accuracy	97.37 %
Incorrectly Classified Instances	2.6294 %
TP Rate	97.9
FP Rate	0.030
Precision	96.2
Recall	97.9
F-Measure	97.7
MCC	94.7
AUC	99.8

#### 4.4.3 R2L Scenario Results

Features with information has gained from 1 to 38 in R2L. Table 4.8 shows that our ALO-MLP classifier in R2L scenario has also reduced the FN and increase the accuracy. Moreover, ALO-MLP has also achieved R2L class of attack in terms of accuracy, incorrect classification, TP, FN, precision, Recall and F-measure. Regarding accuracy, it comes out (98.46%) correct predictions out of 100 total examples). That means our classifier is efficient enough to identifying R2L attack.

FN result show as (0.004%), which is the lower rate. TP rate percentage (99.60%), which considerably good for our mechanism. ALO-MLP has highest precision percentage (98.90%), which means that it has the lower false-positive value. Furthermore, the recall percentage (99.60%).

It has important to note that MMC usually gives lower values which is only percentage (91.10%). Using the ALO-MLP to classify new incoming traffic is very fast, which is the main benefit of this method. However, better performance AUC for testing with percentage (98.00%).

Table 4.8 ALO-MLP Classifier for R2L over metrics.

<b>Evolution</b>	<b>R2L (%)</b>
Accuracy	98.46
Incorrectly Classified Instances	0.1.3594
TP	99.60
FP Rate	0.004
Precision	98.90
Recall	99.60
F-Measure	99.30
MCC	91.10
AUC	98.00

#### 4.4.4 Probes Scenario Results

Observing information gain for features in the network ‘probe’ dataset, remaining ‘38’ features. The feature ‘SRC bytes’ is the most important for successful classification of this traffic class, and its removal causes the most significant increase of misclassification errors. 10-fold cross valuation was used. Table 4.9 shows that our ALO-MLP classifier in probes scenario has also reduced the FN and increase the accuracy as well. Furthermore, ALO -MLP has also achieved probes class of attack in terms of accuracy, incorrect classification, FP, precision, Recall and F-measure. Regarding accuracy, it comes out (98.85%), which means our classifier is efficient enough to identifying probes attack.

False positive results show as (0.109%), which is the lower rate. TP rate percentage (99.80%), which considerably good for our mechanism. ALO-MLP has highest precision percentage (99.90%), which means that it has the lower FN value. Furthermore, the recall percentage (99.80%), which means that it has the lowest false-negative (missed attacks) percentage.

It is important to note that MMC usually gives lower values which is only percentage (92.50%). Using the ALO-MLP to classify new incoming traffic is very fast, which is the main benefit of this method. However, better performance AUC for testing with percentage (99.90%).

Table 4.9 ALO-MLP Classifier for Probe over metrics

Evolution	Probe (%)
Accuracy	98.85
Incorrectly Classified Instances	1.147
TP Rate	99.80
FN Rate	0.109
Precision	99.90
Recall	99.80
F-Measure	99.40
MCC	92.50
AUC	99.90

#### 4.5 Variance Blacklist H-IDPS

As shown in Figure 4.4, the source attack IP “1.1.139.98” has the highest DDoS attack rate in H-IDPS through the NSL-KDD dataset. In fact, H-IDPS prevented this IP from accessing the network 33 times in five minutes during the UDP DDoS attacks. Furthermore, the fewest attack attempts to hit the Apache cloud server were from IP “1.1.139.20”; all IPs shown in Figure 4.5 were classified as blacklist IPs.

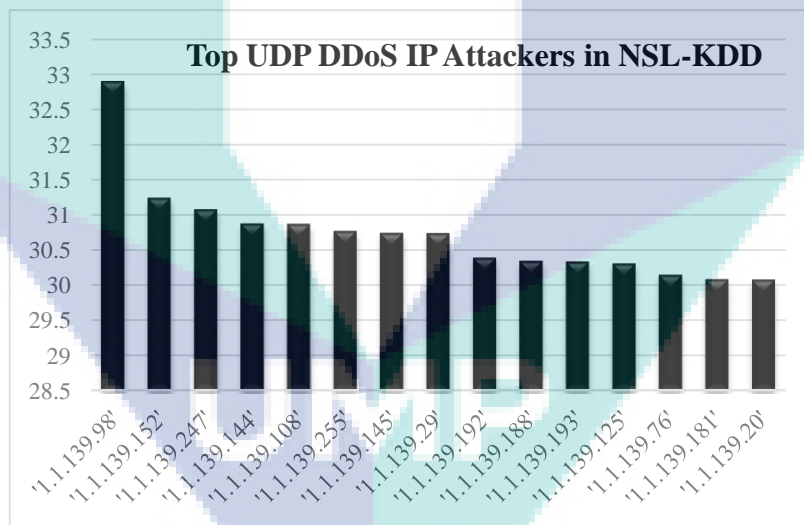


Figure 4.4 UDP highest and lowest IP attack rates.

While, in Figure 4.5 shows the flooding attack rates with the NSL-KDD datasets under the TCP scenario. In this case, address 71.126.222.64 produced the most critical DDoS attack in the cloud server, and this was detected by H-IDPS and sent to the IP blacklist. The fewest attacks in this case came from IP address 69.199.186.70.

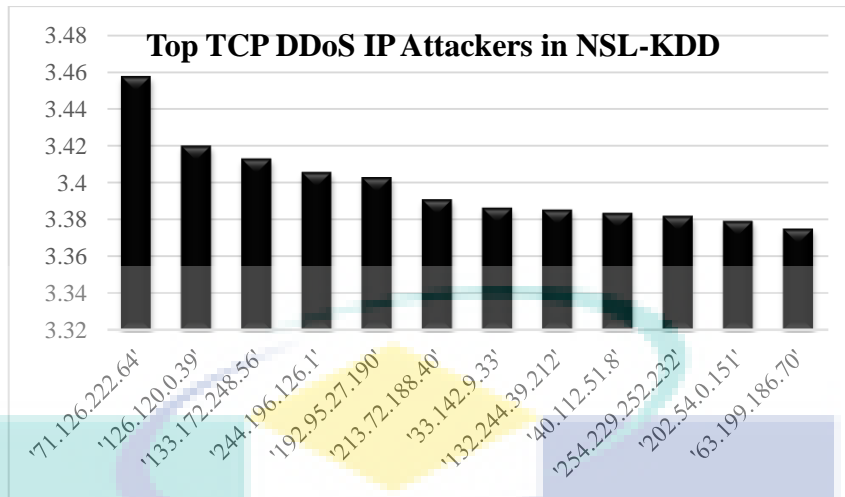


Figure 4.5 TCP highest and lowest IP attack rates.

#### 4.6 Chapter Summary

This Chapter has presented an implementation for preventing DDoS using NSL-KDD dataset. With the current hybrid PCA-LDA, it is challenging to normalize large offline datasets such as NSL-KDD. Then, ALO-MLP classifier has been found to be efficient in detecting and preventing DDoS attacks with high accuracy and low FP rate. H-IDPS blocks any suspicious activity and reports it to the hypervisor, which then evaluates the blocked IPs. However, the results will be further evaluated and compared in the next Chapter.

UMP

## CHAPTER 5

### EVALUATION AND COMPARATIVE ANALYSIS

#### 5.1 Overview

In this Chapter, the proposed mechanism is further analysed and compared with existing mechanisms. The comparison domain covers algorithmic mechanisms as well as other heuristic algorithms. Section 5.2 presents a comprehensive evaluation of the results alongside those from other studies. Section 5.3 evaluates the performance of the ALO-MLP classifier against several common classifiers. Section 5.4 presents a comparative analysis against the results of other mechanisms.

#### 5.2 Evaluation of H-IDPS Snort

To evaluate the Snort H-IDPS in the cloud environment, evaluation examined the Snort pre-processor efficiency, the performance was evaluated using the Packet Header Anomaly Detection (PHAD) pre-processor, Application Layer Anomaly Detection (ALAD) pre-processor, and Learning Rules for Anomaly Detection (LERAD) (Garg & Maheshwari, 2016).

Snort was tested using NSL-KDD traffic and a simulated one-week dataset, with the detected attacks listed day-by-day. The files were downloaded from a local area network. A breakdown of the daily attacks is shown in Figure 5.1. Snort detected 77 out of 180 attacks without using any anomaly-based approaches.

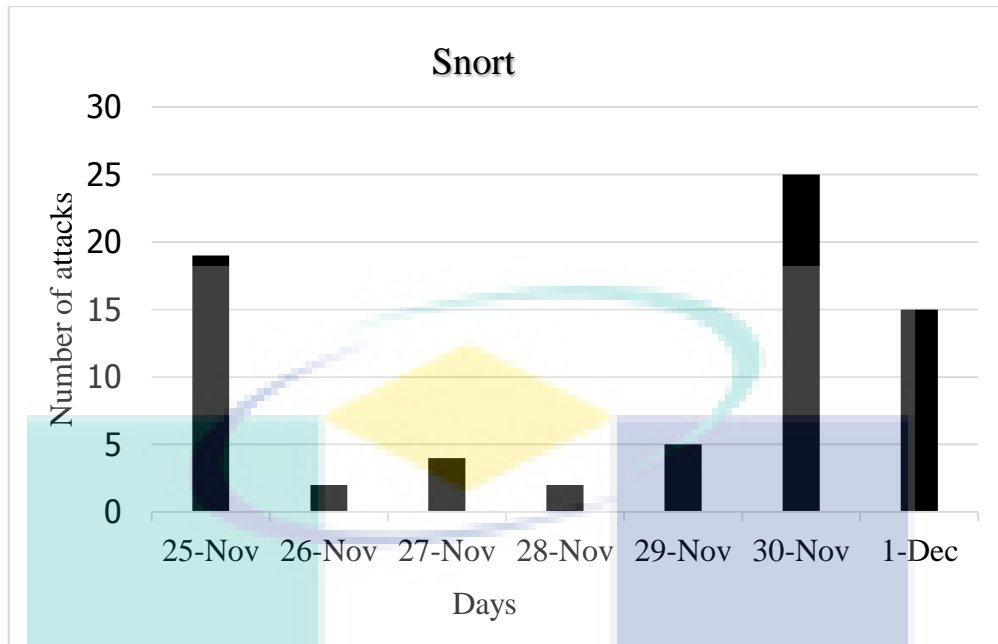


Figure 5.1 Daily DDoS prevention levels for Snort H-IDPS.

### 5.2.1 Snort with ALO-MLP + PHAD

The attacks detected by Snort and PHAD in H-IDPS are shown in Figure 5.2. The addition of PHAD clearly improves the performance of Snort, with the number of attacks detected increasing from 77 to 105. Our implementation of PHAD-C32 processes 2.9 GB of training data and 4.0 GB of test data in 364 s (310 user + 20 VMs), or 95,900 packets per second on a Sparc Ultra 60 with a 450 MHz 64-bit processor, 512 MB memory, and 4 MB cache. The overhead is 23 s of CPU time per simulated day, or 0.026% at the simulation rate. The wall time in our test was 465 s (78% usage), consisting of 165 s of training (77,665 packets per second) and 300 s of testing (73,560 packets per second). The PHAD mechanism uses negligible memory: 34 fields times 32 pairs of 4-byte integers to represent the bounds of each cluster, giving a total of 8 KB. The attacks detected by Snort and PHAD on their own are also shown in Figure 5.2.



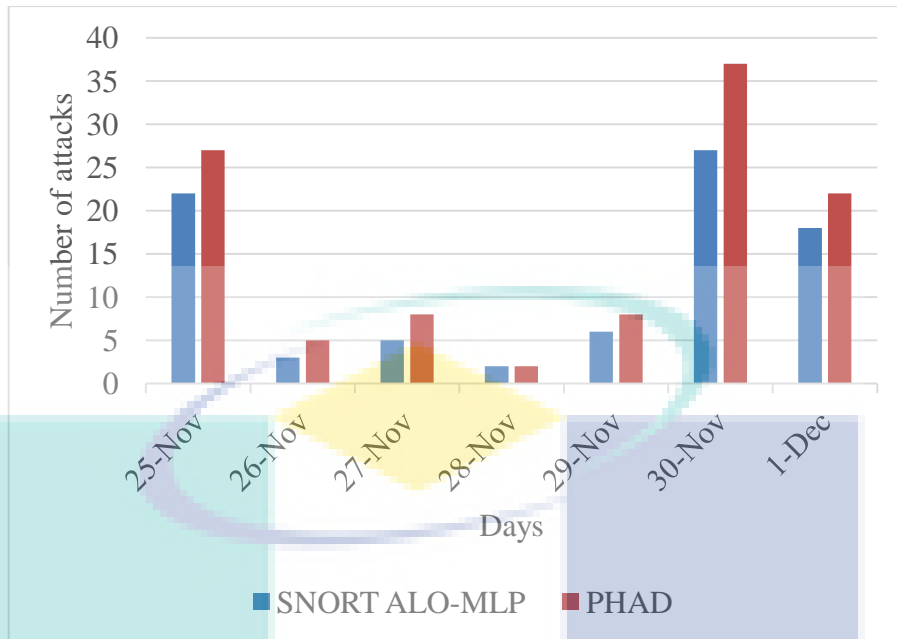


Figure 5.2 Daily DDoS prevention levels for Snort H-IDPS using ALO-MLP with and without PHAD.

### 5.2.2 Snort with ALO-MLP + PHAD + ALAD

Adding both PHAD and ALAD to Snort enables more attacks to be detected, as shown in Figure 5.3. The number of attacks detected increases from 105 to 124 in the Snort + PHAD + ALAD version of H-IDPS, because attacks are detected based on rule definition files. In PHAD and ALAD, attacks are detected using packet headers and the network protocol.

UMP

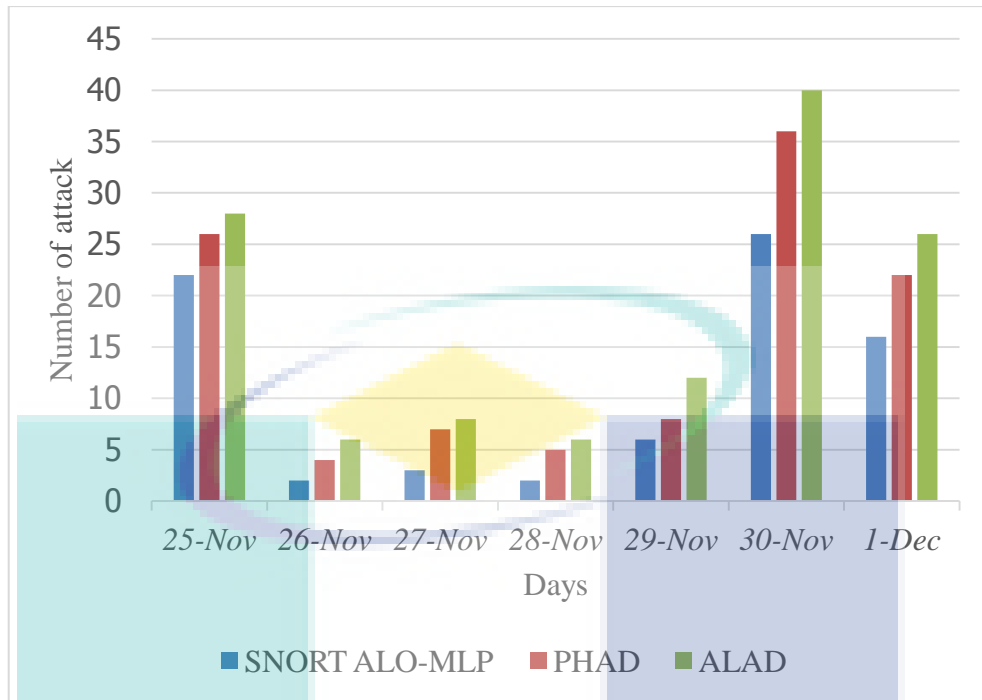


Figure 5.3 Daily DDoS prevention levels for Snort H-IDPS using ALO-MLP compared with PHAD and ALAD.

### 5.2.3 Snort with ALO-MLP+ ALAD + LERAD

The number of attacks detected by Snort + ALAD + LERAD is shown in Figure 5.4. The number of attacks detected has increased from 124 to 149 upon replacing PHAD with LERAD.

UMP

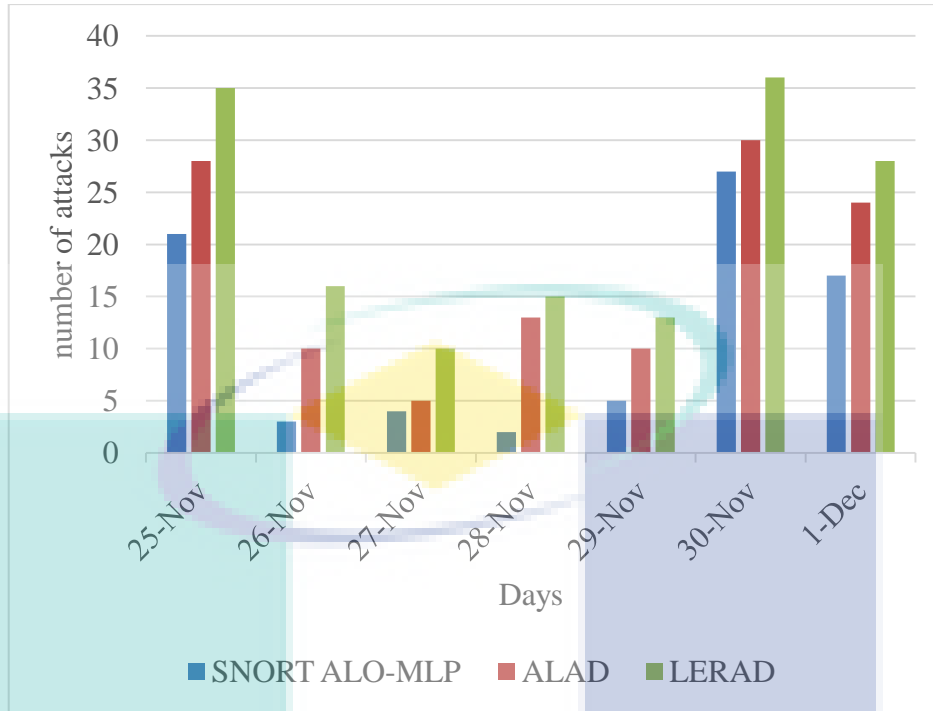


Figure 5.4 Daily DDoS prevention levels for Snort H-IDPS using ALO-MLP classifier compared with ALAD and LERAD.

### 5.3 Evaluation of ALO-MLP in Comparison with Most Common Classifiers

Our classifier was compared with other seven related classifiers, which are MLP, Naive Bayes, K-Mean, Nearest cluster, SVM and Decision tree, NB tree in terms of accuracy for the NSL-KDD attacks seniors. As seen, ALO-MLP classifier is more efficient in locating the optimal solution (optimal solution is found within 10 iterations). Nevertheless, it is observed that our classifier mechanism performs the best (See Appendix B). Figure 5.5, visualise the results of the proposed ALO-MLP with another related classifier. Four attacks of NSL-KDD dataset were simulated and evaluated in term of accuracy for the NSL-KDD classes. In addition, we took our classifier mechanism and applied it attack classes for NSL-KDD starting from DoS, Probing U2R, and ending to R2L.

Naive Bayes Classifier has one main disadvantage that makes a very strong assumption on the shape of data distribution rather than our proposed method. Neuro-Fuzzy experts with 3 datasets, i.e. CAIDA, conficker and UNIANA, but only 6 features. Again, this shows that the proposed H-IDPS using ALO-MLP is better than other ALO-MLP.

K-means is used to detect DDoS attacks and partition large data space effectively, but it has accuracy disadvantage, because It includes dependence on initial centroids, dependence on number of clusters and degeneracy. Our ALO-MLP has overcome these disadvantages of accuracy and achieved the highest rate. The limitation of Nearest Cluster cannot effectively detect U2L and R2L attacks in high accuracy, which means that this one-dimensional distance-based feature representation is not able to well represent the pattern of these two types of attacks.

While, the main disadvantage is SVM can only handle binary-class classification, and for that it shows lower results than our classifier. Decision Tree suffering from overfitting, which is one of cardinal sins in analytics and machine learning and for that it shows less accuracy than our proposed ALO-MLP classifier.

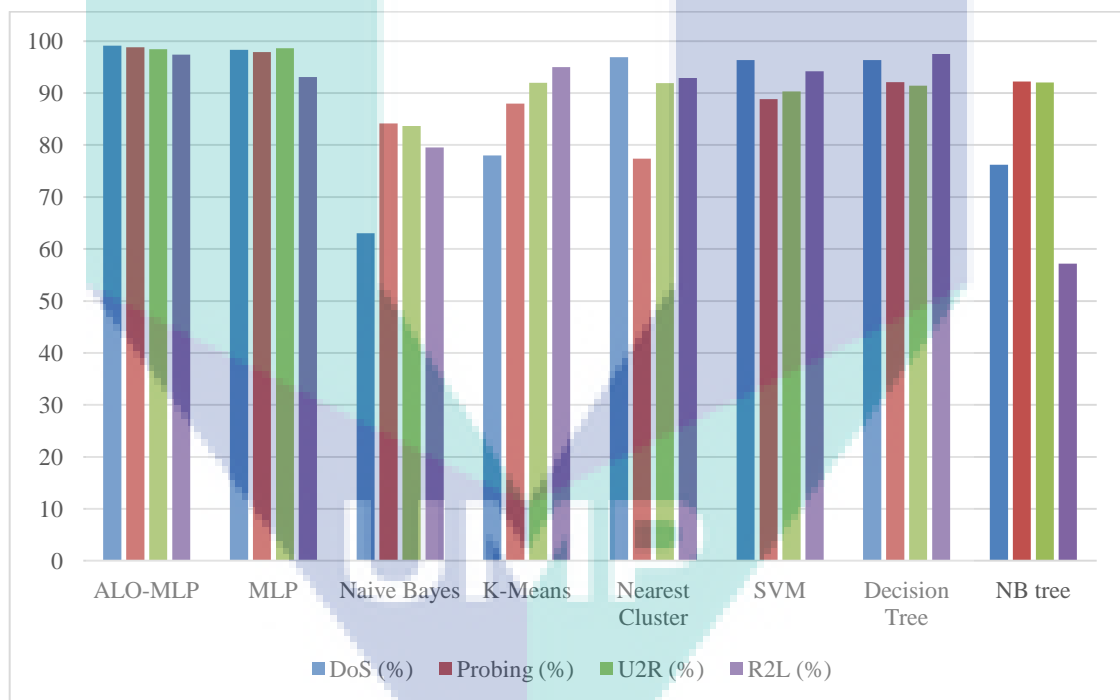


Figure 5.5 ALO-MLP Compared with Most Common Classifiers

## **5.4 Comparative Analysis for ALO-MLP Classifier with Other Classifier Mechanisms**

An evaluation comparison has done with other six related works. These related works have cited in earlier Chapter 1 problem statement. However, we did the comparison using NSL-KDD dataset and eight evaluation metrics as mentioned in Chapter 3 such as: accuracy, incorrect classification rate, FN , TPR , precision, recall and F1 score. DoS, Prob,R2L and U2R DDoS attack classes in NSL-KDD also evaluated using our classifier mechanism which is ALO-MLP and PCA-LDA for the Data pre-processing. After comparison done, accuracy and FN rate is the most important metrics highlighted -which fill the requirement of this thesis objectives .

### **5.4.1 DoS comparison**

Here, NSL-KDD Dos in terms of accuracy, incorrect classification rate, FN, TPR, precision, recall and F1 score were compared. As shown in Table 5.1, our mechanism is 1.1% higher than that of Eid et al. (2011) in terms of accuracy, and 1.1% lower in terms of incorrect classification rate. Furthermore, our mechanism is 0.003 % lower in FN, 0.20% higher in TPR, and 0.70% higher in precision. For the Recall, ALO-MLP mechanism achieved 1.45% higher than their rates. Furthermore, F1 score has achieved 0.80% higher than their mechanism.

As compared to Hassanien et al., (2014) our mechanism is 17.31% higher in terms of accuracy and 4.1% lower in terms of incorrect classification rate. Furthermore, our mechanism is 0.202% lower in FN, 17.96% higher in TPR, and 17.46% higher precision. While the Recall, ALO-MLP mechanism achieved 18.22% higher than their rates. Furthermore, F1 score has achieved 17.51% higher than their mechanism.

Upon comparing our classifier with Emiro De la Hoz et al. (2014), our mechanism is 0.10% higher in terms of accuracy and 2.39% lower in terms of incorrect classification rate. Furthermore, our mechanism is 0.039% lower in FN, 1.12% higher in TPR and 1.96% higher in precision. While the Recall, ALO-MLP mechanism achieved 2.41% higher than their rates. Furthermore, F1 score has achieved 1.71% higher than their mechanism.

The comparison of the proposed system was further evaluated with Pervez and Farid (2014) works. our mechanism is 16.42% higher in terms of accuracy and 4.28% lower in terms of incorrect classification rate. Furthermore, our mechanism is 0.093% lower in FN, 17.65% higher in TPR and 17.61% higher in precision. While the Recall, ALO-MLP mechanism achieved 18.31% higher than their rates. Furthermore, F1 score has achieved 17.90% higher than their mechanism.

For Pajouh et al. (2017), our classifier mechanism is 4.54% higher in terms of accuracy and 1.8% lower in terms of incorrect classification rate. Furthermore, our classifier is 0.063 % lower in FN, 5.42% higher in TPR, and 6.03% higher in precision. Recall has achieved 5.47 rather than their mechanism. However, F1 score has achieved 5.6% higher than their work.

Lastly, our classifier mechanism is also better than developed by Kanakarajan and Muniasamy (2016) In terms of accuracy, our mechanism is 16.71% higher. Meanwhile, our mechanism is 4.59% lower in terms of incorrect classification rate. Furthermore, our mechanism is 0.011% higher in terms of FN, 16.71% higher in terms of TPR, and 17.68% higher in terms of precision. Recall has achieved 17.37 rather than their mechanism. However, F1 score has achieved 16.92% higher than their work



UMP

Table 5.1 DoS comparison with other related works for accuracy, incorrect classification rate, FN , TPR , precision, recall and F1 score.

No.	Reference	Feature selection/ Pre-processing	Classification method	Performance Metrics						
				Accuracy (%)	Incorrect Classification Rate (%)	FN Rate (%)	TPR Rate (%)	Precision (%)	Recall (%)	F1 Score (%)
1	Jaber et al. (2017)	PCA-LDA	ALO-MLP	<u>99.10</u>	0.90	<u>0.05</u>	99.70	99.60	99.70	99.20
2	Eid et al. (2011)	GA-EMD	NB	98.00	2.00	0.08	99.50	98.90	98.25	98.40
3	Hassani en et al., (2014)	PCA	GA-DT	81.97	5.00	0.207	81.74	81.96	81.48	81.69
4	Emiro De la Hoz et al. (2014)	NSGA	GHSOM	87.00	3.60	0.09	78.67	77.15	77.93	77.92
5	Enache and Patriciu (2014)	LOO	OAR-SVM	82.68	5.18	0.098	82.14	81.99	81.39	81.30
6	Pajouh et al. (2017)	LDA	NB-kNNCF	94.56	2.70	0.068	94.28	93.57	93.96	93.66
7	Kanakarajan and Muniassamy (2016)	IG	GAR-forest	82.39	5.49	0.016	82.99	81.92	82.33	82.28

### 5.4.2 Probe comparison

Here, NSL-KDD Probe attack in terms of accuracy, incorrect classification rate, FN, TPR, precision, recall and F1 score were compared. As shown in Table 5.2, our mechanism is 2.9% higher than that of Eid et al. (2011) in terms of accuracy, and 2.7% lower in terms of incorrect classification rate. Furthermore, our mechanism is 0.900 % lower in FN, 11.70% higher in TPR, and 2.8% higher in precision. For the Recall, ALO-MLP mechanism achieved 3.9% higher than their rates. Furthermore, F1 score has achieved 3.4% higher than their mechanism.

As compared to Hassanien et al., (2014) our mechanism is 34.17% higher in terms of accuracy and 33.97% lower in terms of incorrect classification rate. Furthermore, our mechanism is 9.400% lower in FN, 38.76% higher in TPR, and 33.58% higher precision. While the Recall, ALO-MLP mechanism achieved 38.16% higher than their rates. Furthermore, F1 score has achieved 35.19% higher than their mechanism.

Upon comparing our classifier with Enache and Patriciu (2014), our mechanism is 0.12% higher in terms of accuracy and 0.08% lower in terms of incorrect classification rate. Furthermore, our mechanism is 1.863% lower in FN, 1.53% higher in TPR and 0.89% higher in precision. While the Recall, ALO-MLP mechanism achieved 1.02% higher than their rates. Furthermore, F1 score has achieved 1.03% higher than their mechanism.

The comparison of the proposed system was further evaluated with Emiro De la Hoz et al. (2014) works. our mechanism is 3.2% higher in terms of accuracy and 3% lower in terms of incorrect classification rate. Furthermore, our mechanism is 1.829% lower in FN, 4.62% higher in TPR and 3.85% higher in precision. While the Recall, ALO-MLP mechanism achieved 4.62% higher than their rates. Furthermore, F1 score has achieved 3.81% higher than their mechanism.

For Pajouh et al. (2017), our classifier mechanism is 19.04% higher in terms of accuracy and 18.84% lower in terms of incorrect classification rate. Furthermore, our classifier is 1.734 % lower in FN, 20.60% higher in TPR, and 19.70% higher in precision. Recall has achieved 20.30 rather than their mechanism. However, F1 score has achieved 20.29% higher than their work.



Lastly, ALO-MLP mechanism is also better than developed by Kanakarajan and Muniasamy (2016) In terms of accuracy, our mechanism is 20.47% higher. Meanwhile, our mechanism is 20.27% lower in terms of incorrect classification rate. Furthermore, our mechanism is 0.120% higher in terms of FN, 21.69% higher in terms of TPR, and 20.72% higher in terms of precision. Recall has achieved 21.53 rather than their mechanism. However, F1 score has achieved 20.83% higher than their work.

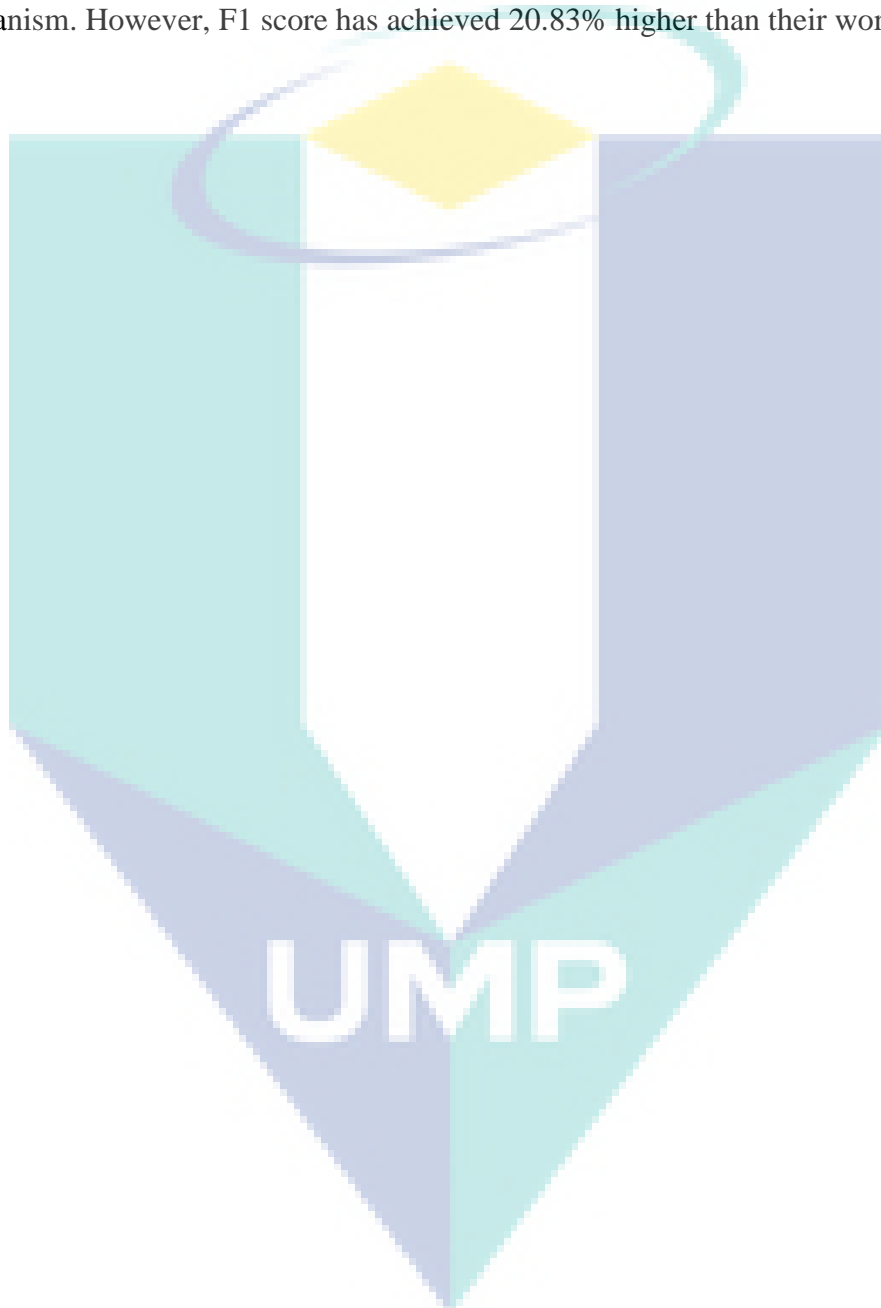


Table 5.2 Probe comparison with other related works for accuracy, incorrect classification rate, FN, TPR, precision, recall and F1 score.

N o.	Referen ce	Feature selection/ Pre-processi ng	Classific ation method	Performance Metrics						
				Accur acy (%)	Incorrec t Classification Rate (%)	FN Rate (%)	TP R Rate (%)	Precis ion (%)	Rec all (%)	F1 Sco re (%)
1	Jaber et al. (2017)	PCA-LDA	ALO-MLP	<u>98.80</u>	1.40	<u>1.90</u> <u>0</u>	99.80	99.00	99.80	99.40
2	Eid et al. (2011)	GA-EMD	NB	95.90	4.10	2.80 0	88.10	96.20	95.90	96.00
3	Hassani en et al., (2014)	PCA	GA-DT	64.63	35.37	11.3 00	61.04	65.42	61.64	64.21
4	Enache and Patriciu (2014)	IG	PSO-SVM	98.68	1.32	0.03 7	98.27	98.11	98.78	98.37
5	Emiro De la Hoz et al. (2014)	NSGA	GHSOM	95.60	4.40	0.07 1	95.18	95.15	95.18	95.59
6	Pajouh et al. (2017)	LDA	NB-kNNCF	79.76	20.24	0.16 6	79.20	79.30	79.50	79.11
7	Kanakar ajan and Muniasa my (2016)	IG	GAR-forest	78.33	21.67	2.02	78.11	78.28	78.27	78.57

### 5.4.3 R2L comparison

Here, NSL-KDD R2L attacks in terms of accuracy, incorrect classification rate, FN, TPR, precision, recall and F1 score were compared. As shown in Table 5.3, our classifier is 4.36% higher than that of Eid et al. (2011) in terms of accuracy, and 4.55% lower in terms of missed detection. Furthermore, our mechanism is 0.392% higher in FN, 2.6% higher in TPR, and 3.5% higher in precision. Recall has achieved 3.4 rather than their mechanism. However, F1 score has achieved 4.8% higher than their work.

As compared to Hassanien et al., (2014) our classifier is 65.56% higher in terms of accuracy and 65.75% lower in terms of incorrect classification rate. Furthermore, our classifier is 1.681% higher in FN, 78.53% higher in TPR, and 81.30% higher precision. Recall has achieved 2.5 rather than their mechanism. However, F1 score has achieved 62.16% higher than their work.

Upon comparing our ALO-MLP with Emiro De la Hoz et al. (2014), ALO-MLP is 4% higher in terms of accuracy and 2.39% lower in terms of missed incorrect classification rate. Furthermore, our ALO-MLP is 1.5% higher in FN, 1.5% higher in TPR and 1.98% higher in precision. Recall has achieved 2.5 rather than their mechanism. However, F1 score has achieved 1.6% higher than their work.

ALO-MLP classifier mechanism is also better than that proposed by Enache and Patriciu (2014). As reported, our ALO-MLP is 10.36% higher in terms of accuracy and 10.55% lower in terms of incorrect classification rate. Furthermore, our ALO-MLP is 0.114% higher in FN, 11.59% higher in TPR, and 14.51% higher in precision. Recall has achieved 14.68% rather than their mechanism. However, F1 score has achieved 8.6% higher than their work.

Also, ALO-MLP is also better than that developed by Pajouh et al. (2017) in terms of accuracy, our ALO-MLP is 4.08% higher. Meanwhile, our ALO-MLP is 4.27% lower in terms of incorrect classification rate. Furthermore, ALO-MLP is 0.389% higher in terms of FN, 5.32% higher in terms of TPR, and 5.49% higher in terms of precision. Recall has achieved 5.05 rather than their mechanism. However, F1 score has achieved 3.19% higher than their work.

Lastly, our ALO-MLP is also better than that developed by Kanakarajan and Muniasamy (2016) In terms of accuracy, our ALO-MLP is 3.7% higher. Meanwhile, ALO-MLP is 2.69% lower in terms of incorrect classification rate. Furthermore, ALO-MLP is 7% higher in terms of FN, 10% higher in terms of TPR, and 2.98% higher in terms of precision. Recall has achieved 2.5 rather than their mechanism. However, F1 score has achieved 1.6% higher than their work.

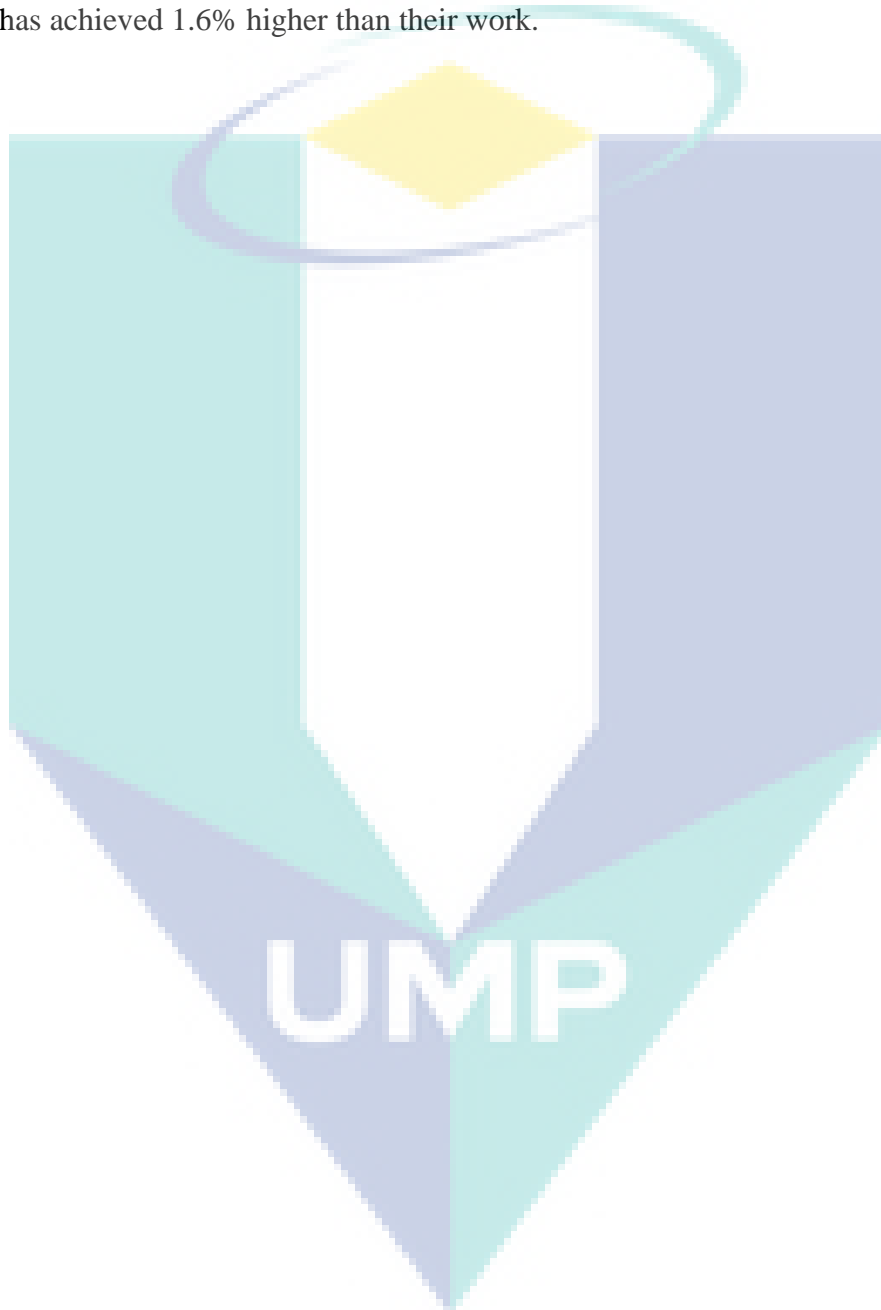


Table 5.3 R2L comparison with other related works for accuracy, incorrect classification rate, FN, TPR, precision, recall and F1 score

No.	Reference	Feature selection/ Pre-processing	Classification method	Performance Metrics						
				Accuracy (%)	Incorrect Classification Rate (%)	FN Rate (%)	TP R Rate (%)	Precision (%)	Recall (%)	F1 Score (%)
1	Jaber et al. (2017)	PCA-LDA	ALO-MLP	<b>98.46</b>	1.35	<b>0.400</b>	99.60	99.60	99.30	91.10
2	Eid et al. (2011)	GA and EMD	NB	94.1	5.90	0.08	97.00	96.10	95.90	95.90
3	Hassanien et al., (2014)	PCA	GA-DT	32.90	67.10	2.081	21.07	24.30	18.00	28.94
4	Enache and Patriciu (2014)	IG	PSO-SVM	88.10	11.90	0.286	88.01	85.09	84.62	82.62
5	Emiro De la Hoz et al. (2014)	NSGA	GHSOM	94.38	5.62	0.011	94.28	94.11	94.25	94.29
6	Pajouh et al. (2017)	LDA	NB-kNNCF	84.68	15.32	0.169	84.19	84.12	84.47	84.52
7	Kanakarajan and Muniassamy (2016)	IG	GAR-forest	78.98	27.02	78.15	78.88	78.10	78.22	78.55

#### 5.4.4 U2R comparison

Here, NSL-KDD Dos in terms of accuracy, incorrect classification rate, FN, TPR, precision, recall and F1 score also were compared. As shown in Table 5.4, our classifier mechanism is 1.97% higher than that of Eid et al. (2011) in terms of accuracy, and 2% lower in terms of missed detection. Furthermore, our ALO-MLP is 0.002% higher in FN, 0.9% higher in TPR, and 0.5% higher in precision. Recall has achieved 2.5 rather than their mechanism. However, F1 score has achieved 1.6% higher than their work.

As compared to Hassanien et al., (2014) our mechanism is 29.02% higher in terms of accuracy and 33.05% lower in terms of incorrect classification rate. Furthermore, our classifier is 2.677% higher in FN, 31.62% higher in TPR, and 30.87% higher precision Recall has achieved 29.57 rather than their mechanism. However, F1 score has achieved 28.87% higher than their mechanism.

Upon comparing our ALO-MLP with Emiro De la Hoz et al. (2014), our proposed mechanism is 4.37% higher in terms of accuracy and 4.4% lower in terms of missed incorrect classification rate. Furthermore, our classifier mechanism is 0.006% higher in FN, 4.9% higher in TPR and 6.26% higher in precision. While, in Recall has achieved 8.24 rather than their mechanism. Nevertheless, F1 score has achieved 7.83% higher than their mechanisms.

Our mechanism is also better than that proposed by Pajouh et al. (2017) as reported, ALO-MLP is 30.21% higher in terms of accuracy and 21.24% lower in terms of incorrect classification rate. Furthermore, ALO-MLP is 11.015% higher in FN, 30.72% higher in TPR, and 29.19% higher in precision. While, in Recall has achieved 30.82 rather than their mechanism. Nevertheless, F1 score has achieved 29.98% higher than their mechanism.

Also, our classifier mechanism is also better than that developed by Rastegari et al. (2015) In terms of accuracy, our ALO-MLP is 17.46% higher. Meanwhile, ALO-MLP is 17.49% lower in terms of incorrect classification rate. Furthermore, ALO-MLP is 8.867% higher in terms of FN, 18.4% higher in terms of TPR, and 16.62% higher in terms of precision. Recall has achieved 18.02% rather than their mechanism. Nevertheless, F1 score has achieved 17.79% higher than their mechanism.

Lastly, our mechanism is also better than that developed by Kanakarajan and Muniasamy (2016) In terms of accuracy, ALO-MLP is 19.81% higher. Meanwhile, our ALO-MLP is 2.25% lower in terms of incorrect classification rate. Furthermore, ALO-MLP is 12.578% higher in terms of FN, 20.82% higher in terms of TPR, and 18.8% higher in terms of precision. Recall has achieved 20.71% rather than their mechanism. Nevertheless, F1 score has achieved 19.84% higher than their mechanism.

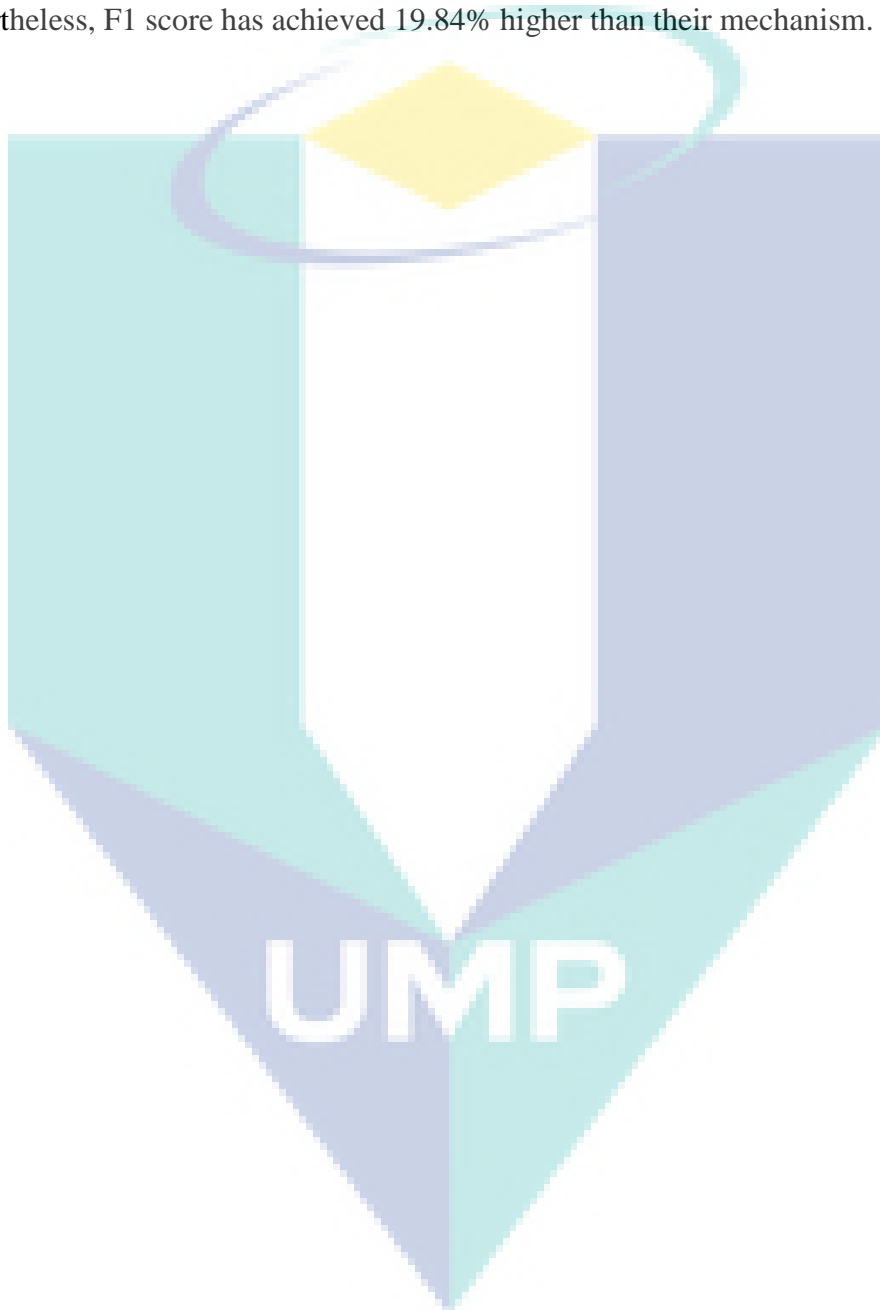


Table 5.4 U2R comparison with other related works for accuracy, incorrect classification rate, FN, TPR, precision, recall and F1 score

No.	Reference	Feature selection/Pre-processing	Classification method	Performance Metrics						
				Accuracy (%)	Incorrect Classification Rate (%)	FN Rate (%)	TPR Rate (%)	Precision (%)	Recall (%)	F1 Score (%)
1	Jaber et al. (2017)	PCA-LDA	ALO-MLP	<b>97.37</b>	02.60	<b>0.003</b>	97.90	96.20	97.90	97.00
2	Eid et al. (2011)	GA and EMD	NB	95.40	4.60	0.005	97.00	95.70	95.40	95.40
3	Hassanien et al., 2014	PCA	GA-DT	68.35	35.65	2.680	66.28	65.33	68.33	68.13
4	Emiro De la Hoz et al. (2014)	NSGA	GHSOM	93.00	07.00	0.009	93.00	89.94	89.66	89.17
5	Pajouh et al. (2017)	LDA	NB-kNNCF	67.16	23.84	11.018	67.18	67.01	67.08	67.02
6	Rastegari et al. (2015)	CFS	GA classifier	79.91	20.09	8.690	79.50	79.58	79.88	79.21
7	Kanakarajan and Muniasamy (2016)	IG	GAR-forest	77.56	0.35	12.581	77.08	77.40	77.19	77.16



## 5.5 Chapter Summary

The results presented in this Chapter demonstrate that the current mechanism offers high accuracy and efficiency for H-IDPS in the cloud hypervisor environment. For the purposes of comparison, the mechanism was split into three parts: PCA, LDA, and the proposed method with the ALO-MLP classifier. The proposed mechanism was then compared with existing mechanisms in terms of the detection and prevention of DDoS attacks. Further, a feasible estimation of the IPs observed in the NSL-KDD blacklist was able to identify the IP that was attacking the victim, i.e. the cloud environment. Thus, the proposed mechanism has addressed the research problems outlined in this thesis. The next Chapter discusses the advantages and limitations of the current mechanism.

The logo for UIMP (Universiti Malaysia Perlis) is a large, downward-pointing arrow shape. It is composed of four triangular sections meeting at a central point. The top-left and bottom-right sections are light blue, while the top-right and bottom-left sections are a slightly darker shade of blue. The letters 'UIMP' are written in a bold, white, sans-serif font across the center of the arrow.

UIMP

## CHAPTER 6

### CONCLUSION

#### 6.1 Overview

DDoS makes use of many machines distributed in the environment to attack the services on the web. Therefore, it is crucial to detect the attack as early as possible. H-IDPS are necessary to provide the desired protection against the attack. Therefore, it is essential to combine these technologies and install them in one device. In an ideal situation, the device will guarantee accuracy of true positives at its maximum level in detecting the threats and reducing the number of FN. Nevertheless, it is interesting to note that the number of FN has been reduced remarkably due to the improvement in the current technology. Meanwhile, H-IDPS can block the immediate attacks such as viruses and worms and certain new threats.

#### 6.2 Contribution

The main contribution of this thesis is the design a new classifier mechanism for detecting and preventing DDoS attack in cloud computing environment. The development work of the new mechanism was executed in three phases.

**Phase 1:** A newly designed classifier mechanism based on ALO-MLP for H-IDPS in cloud environment. The main privilege of using ALO is to feed the weights of MLP, and for that will produce a better performance mechanism as a classifier which will led to better H-IDPS accuracy and less FN.

**Phase 2:** The development of ALO-MLP classifier mechanism with Snort and Weka produced a robust classifier against DDoS attack using NSL-KDD traffics. Data

analysis and pattern recognition, we were able to build another high-level DDoS prevention mechanism.

**Phase 3:** Performance evaluation of the ALO-MLP related mechanisms. These mechanisms can neither fulfil all the requirements proposed nor depict an architecture that is suitable for the protection of cloud computing environment due to the low of ty accuracy and high of False alarm rate as seen in Chapter 2. Therefore, ALO-MLP classifier shown the highest performance above them using machine learning metrics. Furthermore, a comparison with these mechanisms based on four attack classes of NSL-KDD, which are DoS, R2L, U2R and Probe. Thus, we did a comparison with each attack classes in term of accuracy, incorrect classification rate, FN, TPR, precision, recall and F1 score. However, all these metrics shows the highest performance in Chapter 5. more precisely, the accuracy and FN. For the DoS class scenario our ALO-MLP classifier achieve 99.10% in term of accuracy, and 0.005% in FN. In Probe scenario has achieved 98.80% in accuracy, and 1.40% in FN. For the R2L accuracy, it shown 98.46% and 0.400 in FN. While, U2R shown 0.003% in accuracy and FN. As a result, all these three contributions have achieved the thesis objectives requirement and led to prevent the DDoS attack in lower FN and higher accuracy.

### 6.3 Future Works

There are several methods which are related to H-IDPS, but it is in rear class for the H-IDPS cloud. These methods are generic algorithm, neural networks, and fuzzy theory in H-IDPS. As far as the study in future is concerned it would be an open research by adding ALO-MLP with Harmony based strategies.

## REFERENCES

- Aburomman, A. A., & Reaz, M. B. I. (2016). *Ensemble of Binary SVM Classifiers Based on PCA and LDA Feature Extraction for Intrusion detection*. Advanced Information Management, Communicates, Electronic and Automation Control Conference , pp. 636-640.IEEE.
- Agrawal, N., & Tapaswi, S. (2017). Defense Schemes for Variants of Distributed Denial-Of-Service (DDoS) attacks in cloud computing: A survey. *Information Security Journal: A Global Perspective*, 26(2), pp. 61-73.
- Ahmad, T., Haque, M. A., Al-Nafjan, K., & Ansari, A. A. (2013). Development of Cloud Computing and Security Issues. *Information and Knowledge Management*. Vol. 3, No. 1.
- Badis, H., Doyen, G., & Khatoun, R. (2014). *Toward a Source Detection of Botclouds: a PCA-Based Approach*. International Conference on Autonomous Infrastructure, Management and Security, pp. 105-117. Springer, Berlin, Heidelberg.
- Behal, S., & Kumar, K. (2016). Trends in Validation of DDoS Research. *Procedia Computer Science*, 85, pp.7-15.
- Bhardwaj, A., Subrahmanyam, G. V. B., Avasthi, V., Sastry, H., & Goundar, S. (2016). *DDoS attacks, New DDoS Taxonomy and Mitigation Solutions—A survey*. International Conference on Signal Processing, Communication, Power and Embedded System, pp.793-798. IEEE.
- Bharot, N., Verma, P., Sharma, S., & Suraparaju, V. (2017). Distributed Denial-Of-Service Attack Detection and Mitigation Using Feature Selection and Intensive Care Request Processing Unit. *Arabian Journal for Science and Engineering*, 43(2), pp.959-967.
- Bhat, A. H., Patra, S., & Jena, D. (2013). Machine Learning Approach For Intrusion Detection on Cloud Virtual Machines. *International Journal of Application or Innovation in Engineering & Management (IJAIEEM)*, 2(6), pp.56-66.
- Birje, M. N., Challagidad, P. S., Goudar, R., & Tapale, M. T. (2017). Cloud computing Review: Concepts, Technology, Challenges and Security. *International Journal of Cloud Computing*, 6(1), pp.32-57.
- Boughorbel, S., Jarray, F., & El-Anbari, M. (2017). Optimal Classifier for Imbalanced Data Using Matthews Correlation Coefficient metric. *PLoS ONE*, 12(6), pp.1-17.
- Chatterjee, T., & Bhattacharya, A. (2014). VHDL Modeling of Intrusion Detection & Prevention System (IDPS) A Neural Network Approach. *International Journal of Computer Trends and Technology (IJCTT)*. 8(1), pp.52-56.

- Chen, Chia-Mei, D. J. Guan, Yu-Zhi Huang, and Ya-Hui Ou. *Attack Sequence Detection in Cloud Using Hidden Markov Model*. 7<sup>th</sup> Asia Joint Conference on Information Security, pp. 100-103. IEEE.
- Chiba, Z., Abghour, N., Moussaid, K., & Rida, M. (2016). A Cooperative and Hybrid Network Intrusion Detection Framework in Cloud Computing Based on Snort And Optimized Back Propagation Neural Network. *Procedia Computer Science*, 83, pp. 1200-1206.
- Cunningham, J. P., & Ghahramani, Z. (2015). Linear Dimensionality Reduction: Survey, Insights, and Generalizations. *The Journal of Machine Learning Research*, 16(1), pp. 2859-2900.
- Da Silva Filho, H. C., de Figueiredo Carneiro, G., Costa, E. S. M., & Monteiro, M. (2018). Tools to Support SMEs to Migrate to the Cloud: Opportunities and Challenges. In *Information Technology-New Generations*, pp. 159-165. Springer, Cham.
- Daryabar, F., Dehghantanha, A., Eterovic-Soric, B., & Choo, K.-K. R. (2016). Forensic Investigation of OneDrive, Box, GoogleDrive and Dropbox Applications on Android and iOS devices. *Australian Journal of Forensic Sciences*, 48(6), pp. 615-642.
- De la Hoz, E., de la Hoz, E., Ortiz, A., Ortega, J., & Martínez-Álvarez, A. (2014). Feature Selection by Multi-Objective Optimisation: Application to Network Anomaly Detection by Hierarchical Self-Organising Maps. *Knowledge-Based Systems*, 71, pp. 322-338.
- De la Hoz, E., De La Hoz, E., Ortiz, A., Ortega, J., & Prieto, B. (2015). PCA Filtering and Probabilistic SOM for Network Intrusion Detection. *Neurocomputing*, 164, pp. 71-81.
- Denning, T., Kohno, T., & Shostack, A. (2013). Control-Alt-Hack: A card game for computer security outreach and education. Technical Symposium on Computer Science Education, pp. 729-729. ACM.
- Deshpande, P., Sharma, S. C., Peddoju, S. K., & Junaid, S. (2014). HIDS: A Host Based Intrusion Detection System for Cloud Computing Environment. *International Journal of System Assurance Engineering and Management*, 9(3), pp. 567-576.
- Drago, I., Mellia, M., M Munafo, M., Sperotto, A., Sadre, R., & Pras, A. (2012). *Inside Dropbox: Understanding Personal Cloud Storage Services*. Internet Measurement Conference, pp. 481-494. ACM.
- Eid, H. F., Darwish, A., Hassanien, A. E., & Kim, T.-h. (2011). Intelligent Hybrid Anomaly Network Intrusion Detection System. In *Communication and Networking*, pp. 209-218. Springer.

- Elkhadir, Z., Chougali, K., & Benattou, M. (2017, April). *A Median Nearest Neighbors LDA for Anomaly Network Detection*. International Conference on Codes, Cryptology, and Information Security, pp. 128-141. Springer, Cham.
- Enache, A. C., & Patriciu, V. V. (2014, May). *Intrusions Detection Based On Support Vector Machine Optimized with Swarm Intelligence*. IEEE International Symposium on Applied Computational Intelligence and Informatics, pp. 153-158. IEEE.
- Everett, C. (2009). Cloud computing – A Question of Trust. *Computer Fraud & Security*, 2009(6), pp. 5-7.
- Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges. *Computers & Security*, 28(1), pp.18-28.
- Garg, A., & Maheshwari, P. (2016, January). *PHAD: Packet Header Anomaly Detection*. International Conference on Intelligent Systems and Control (ISCO), pp. 1-5. IEEE.
- Ghamisi, P., & Benediktsson, J. A. (2015). Feature Selection Based on Hybridization Of Genetic Algorithm and Particle Swarm Optimization. *IEEE Geoscience and Remote Sensing Letters*, 12(2), pp. 309-313.
- Ghosh, P., & Mitra, R. (2015, February). *Proposed GA-BFSS and Logistic Regression Based Intrusion Detection System*. Conference on Computer, Communication, Control and Information Technology, pp. 1-6. IEEE.
- Gillman, D., Lin, Y., Maggs, B., & Sitaraman, R. K. (2015). Protecting Websites from Attack with Secure Delivery Networks. *Computer*, 48(4), pp. 26-34.
- Grossman, R. L., Gu, Y., Sabala, M., & Zhang, W. (2009). Compute and Storage Clouds Using Wide Area High Performance Networks. *Future Generation Computer Systems*, 25(2), pp.179-183.
- Hassanien, A. E., Kim, T.-H., Kacprzyk, J., & Awad, A. I. (2014). *Bio-inspiring Cyber Security and Cloud Services: Trends and Innovations*, vol. 70. Springer.
- Hatef, M. A., Shaker, V., Jabbarpour, M. R., Jung, J., & Zarrabi, H. (2018). HIDCC: A Hybrid Intrusion Detection Approach in Cloud Computing. *Concurrency and Computation: Practice and Experience*, 30(3).
- Himmel, M. A., & Grossman, F. (2014). Security on Distributed Systems: Cloud Security Versus Traditional IT. *IBM Journal of Research and Development*, 58(1), pp. 3-1.
- Hota, H., & Shrivastava, A. K. (2014). *Data Mining Approach for Developing Various Models Based on Types of Attack and Feature Selection as Intrusion Detection Systems (IDS)*. International Conference on Advanced Computing, Networking, and Informatics, vol 243, pp. 845-851. Springer.



- Ingre, B., & Yadav, A. (2015). *Performance Analysis of NSL-KDD Dataset using ANN*. International Conference on Signal Processing and Communication Engineering Systems, pp. 92-96. IEEE.
- Iyengar, N. C. S., Banerjee, A., & Ganapathy, G. (2014). A Fuzzy Logic Based Defense Mechanism Against Distributed Denial of Services Attack in Cloud Environment. *International Journal of Communication Networks and Information Security*, 6(3).
- Jaber, A. N., Zolkipli, M. F., Shakir, H. A., & Jassim, M. R. (2017). *Host Based Intrusion Detection and Prevention Model Against DDoS Attack in Cloud Computing*. International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 241-252. Springer, Cham.
- Jiao, J., Ye, B., Zhao, Y., Stones, R. J., Wang, G., Liu, X. & Xie, G. (2017). *Detecting TCP-Based DDoS Attacks in Baidu Cloud Computing Data Centers*. 2017 IEEE Symposium on Reliable Distributed Systems, pp. 256-258. IEEE.
- Kamatchi, R., Ambekar, K., & Parikh, Y. (2017). Security Mapping of a Usage Based Cloud System. *Network Protocols and Algorithms*, 8(4), pp. 56-71.
- Kanakarajan, N. K., & Muniasamy, K. (2016). *Improving the Accuracy of Intrusion Detection Using GAR-Forest with Feature Selection*. International Conference on Frontiers in Intelligent Computing: Theory and Applications, pp. 539-547. Springer, New Delhi.
- Karimi, A. M., Niyaz, Q., Sun, W., Javaid, A. Y., & Devabhaktuni, V. K. (2016). *Distributed Network Traffic Feature Extraction for A Real-Time IDS*. International Conference on Electro Information Technology, pp. 0522-0526. IEEE.
- Kaul, S., Sood, K., & Jain, A. (2017). Cloud Computing and its Emerging Need: Advantages and Issues. *International Journal of Advanced Research in Computer Science*, 8(3).
- Kazemi, S., Aghazarian, V., & Hedayati, A. (2015). Improving False Negative Rate in Hypervisor-Based Intrusion Detection in IaaS Cloud. *International Journal of Computing and Technology*.2(9).
- Keerthi Vasan, K., & Surendiran, B. (2016). Dimensionality reduction using Principal Component Analysis for network intrusion detection. *Perspectives in Science*, 8, pp. 510-512.
- Khan, M. T., Hyun, M., Kanich, C., & Ur, B. (2018). Forgotten But Not Gone: Identifying the Need for Longitudinal Data Management in Cloud Storage. Conference on Human Factors in Computing Systems (p. 543). ACM.
- Kim, H., Kim, J., Kim, Y., Kim, I., & Kim, K. J. (2018). Design of Network Threat Detection and Classification Based on Machine Learning on Cloud Computing. *Cluster Computing*, 1-10.

- Kizza, J. M. (2017). System Intrusion Detection and Prevention. In *Guide to computer network security*, pp. 275-301. Springer.
- Kritikos, K., Kirkham, T., Kryza, B., & Massonet, P. (2017). Towards a Security-Enhanced PaaS Platform for Multi-Cloud Applications. *Future Generation Computer Systems*, 67, pp.206-226.
- Kumar, P. A. R., & Selvakumar, S. (2013). Detection Of Distributed Denial of Service Attacks using an Ensemble Of Adaptive and Hybrid Neuro-Fuzzy Systems. *Computer Communications*, 36(3), pp.303-319.
- Latha, S., & Prakash, S. J. (2017). *A Survey on Network Attacks and Intrusion detection Systems*. International Conference on Advanced Computing and Communication Systems, pp.1-7. IEEE
- Lee, Y.-J., Yeh, Y.-R., & Wang, Y.-C. F. (2013). Anomaly Detection via Online Oversampling Principal Component Analysis. *IEEE Transactions on Knowledge and Data Engineering*, 25(7), pp.1460-1470.
- Li, H., Liu, B., Mukherjee, A., & Shao, J. (2014). Spotting Fake Reviews Using Positive-Unlabeled Learning. *Computación y Sistemas*, 18(3), pp.467-475.
- Lin, Y., & Li, B. (2018). WebAD<sup>2</sup>: A Cascading Model Based on Machine Learning for Web Attacks Detection. International Conference on Security and Privacy in Communication Systems, pp. 145-165. Springer International Publishing.
- Lonea, A. M., Popescu, D. E., & Tianfield, H. (2013). Detecting DDoS Attacks In Cloud Computing Environment. *International Journal of Computers Communications & Control*, 8(1), pp.70-78.
- Maher, M., Smith, A., & Margiotta, J. (2014). *A Synopsis of the Defense Advanced Research Projects Agency (DARPA) Investment in Additive Manufacture and What Challenges Remain*. International Society for Optics and Photonics, vol. 8970, pp. 897002.
- Mani, M., Bozorg-Haddad, O., & Chu, X. (2018). Ant Lion Optimizer (ALO) Algorithm. In *Advanced Optimization by Nature-Inspired Algorithms*, pp. 105-116. Springer.
- Manickam, M., & Rajagopalan, S. P. (2018). A hybrid multi-Layer Intrusion Detection System in Cloud. *Cluster Computing*, pp.1-9.
- Mirjalili, S. (2015). The Ant Lion Optimizer. *Advances in Engineering Software*, 83, pp. 80-98.
- Mkuzangwe, N. N. P., & Nelwamondo, F. V. (2017). *A Fuzzy Logic Based Network Intrusion Detection System for Predicting the TCP SYN Flooding Attack*. Asian Conference on Intelligent Information and Database Systems, pp. 14-22. Springer, Cham.

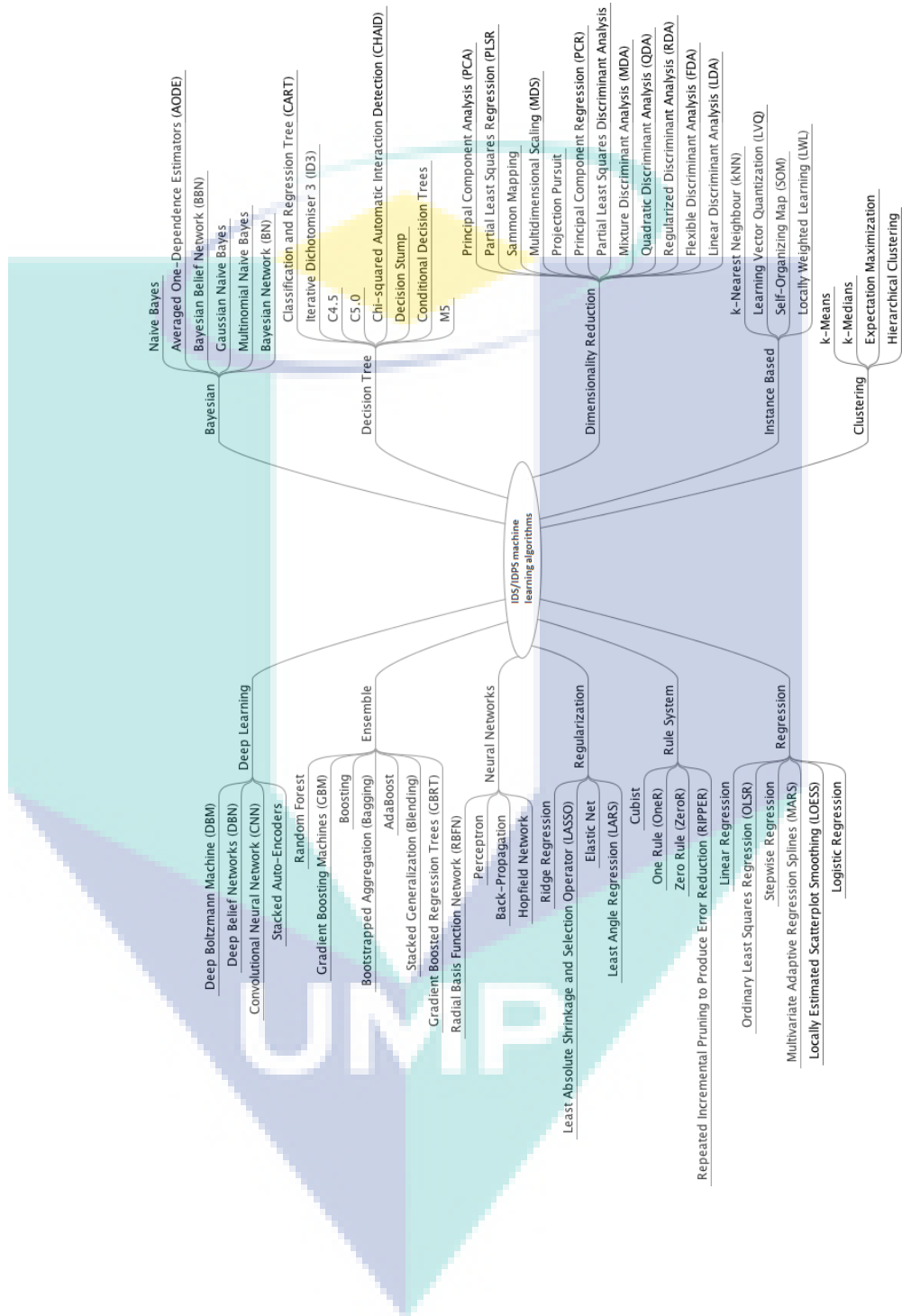


- Modi, C. N., & Acha, K. (2017). Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review. *the Journal of Supercomputing*, 73(3), 1192-1234.
- Mohamed, E. M., Abdelkader, H. S., & El-Etriby, S. (2012). Enhanced data security model for cloud computing. International Conference on Informatics and Systems (pp. CC-12). IEEE.
- More, S., & Chaudhari, S. (2016). Third Party Public Auditing Scheme for Cloud Storage. *Procedia Computer Science*, 79, pp. 69-76.
- Mukhopadhyay, I., Chakraborty, M., Chakrabarti, S., & Chatterjee, T. (2011). Back Propagation Neural Network Approach to Intrusion Detection System. International Conference on Recent Trends in Information Systems, pp. 303-308. IEEE.
- Nazer, G. M., & Selvakumar, A. A. L. (2011). Current Intrusion Detection Techniques in Information Technology-A Detailed Analysis. *European Journal of Scientific Research*, 65(4), pp. 611-624.
- Ndibwile, J. D., Govardhan, A., Okada, K., & Kadobayashi, Y. (2015). *Web Server Protection Against Application Layer DDoS Attacks Using Machine Learning and Traffic Authentication*. Annual Computer Software and Applications Conference, vol. 3, pp. 261-267. IEEE.
- Osanaiye, O., Cai, H., Choo, K.-K. R., Dehghantanha, A., Xu, Z., & Dlodlo, M. (2016). Ensemble-Based Multi-Filter Feature Selection Method for DDoS Detection in Cloud Computing. *EURASIP Journal on Wireless Communications and Networking*, 2016(1), pp.130.
- Pajouh, H. H., Dastghaibyfar, G., & Hashemi, S. (2017). Two-Tier Network Anomaly Detection Model: A Machine Learning Approach. *Journal of Intelligent Information Systems*, 48(1), pp. 61-74.
- Patel, A., Taghavi, M., Bakhtiyari, K., & JúNior, J. C. (2013). An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of Network and Computer Applications*, 36(1), 25-41.
- Perez-Botero, D., Szefer, J., & Lee, R. B. (2013). *Characterizing Hypervisor Vulnerabilities in Cloud Computing Servers*. International workshop on Security in cloud computing, pp. 3-10. ACM.
- Pervez, M. S., & Farid, D. M. (2014). *Feature Selection and Intrusion Classification in NSL-KDD Cup 99 Dataset Employing SVMs*. 8th International Conference on Software, Knowledge, Information Management and Applications, pp. 1-6. IEEE.
- Purwanto, Y., & Rahardjo, B. (2014). *Traffic Anomaly Detection in DDoS Flooding Attack*. International Conference on Telecommunication Systems Services and Applications, pp. 1-6. IEEE.

- Raja, S., & Ramaiah, S. (2016). Performance Comparison of Neuro-Fuzzy Cloud Intrusion Detection Systems. *Int. Arab J. Inf. Technol.*, 13(1A), pp.142-149.
- Rastegari, S., Hingston, P., & Lam, C.-P. (2015). Evolving statistical Rulesets For Network Intrusion Detection. *Applied Soft Computing*, 33, pp. 348-359.
- Rathore, M. M., Ahmad, A., & Paul, A. (2016). Real Time Intrusion Detection System for Ultra-High-Speed Big Data Environments. *The Journal of Supercomputing*, 72(9), pp. 3489-3510.
- Reddy, N. C. S., Vemuri, P. C. R., & Govardhan, A. (2017). Evaluation of PCA and K-means Algorithm for Efficient Intrusion Detection. *International Journal of Applied Engineering Research*, 12(12), pp. 3370-3376.
- Saad, A. A., Khalid, C., & Mohamed, J. (2015). *Network Intrusion Detection System Based on Direct LDA*. 2015 Third World Conference on Complex Systems, pp. 1-6. IEEE.
- Sahani, R., Rout, C., Badajena, J. C., Jena, A. K., & Das, H. (2018). Classification of Intrusion Detection Using Data Mining Techniques. *In Progress in Computing, Analytics and Networking*, pp. 753-764. Springer, Singapore.
- Saied, A., Overill, R. E., & Radzik, T. (2016). Detection Of Known and Unknown DDoS Attacks Using Artificial Neural Networks. *Neurocomputing*, 172, pp. 385-393.
- Salman, T., Bhamare, D., Erbad, A., Jain, R., & Samaka, M. (2017). Machine Learning for Anomaly Detection and Categorization in Multi-Cloud Environments. *International Conference on Cyber Security and Cloud Computing*, pp. 97-103. IEEE.
- Sanchez, M. A., Castillo, O., & Castro, J. R. (2017). An Overview of Granular Computing Using Fuzzy Logic Systems. *In Nature-Inspired Design of Hybrid Intelligent Systems*, pp. 19-38. Springer, Cham.
- Shah, B., & Trivedi, B. H. (2015). *Reducing Features of KDD CUP 1999 Dataset for Anomaly Detection using Back Propagation Neural Network*. *International Conference on Advanced Computing & Communication Technologies*. pp. 247-251. IEEE.
- Singh, J., Kumar, K., Sachdeva, M., & Sidhu, N. (2012). DDoS Attack's Simulation using Legitimate and Attack Real Data Sets. *International Journal of Scientific & engineering research*, 3(6), pp.1-5.
- Singh, K., Singh, P., & Kumar, K. (2017). Application Layer HTTP-GET Flood DDoS Attacks: Research landscape and challenges. *Computers & Security*, 65, pp. 344-372.

- Su, B., Ding, X., Wang, H., & Wu, Y. (2017). Discriminative Dimensionality Reduction For Multi-Dimensional Sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PP(99)*, pp.1-1.
- Subashini, S., & Kavitha, V. (2011). A Survey on Security Issues in Service Delivery Models of Cloud Computing. *Journal of Network and Computer Applications*, *34(1)*, pp.1-11.
- Sudharsan, N. S., & Latha, K. (2013). *Improvising Seeker Satisfaction in Cloud Community Portal: Dropbox*. 2013 International Conference on Communication and Signal Processing, pp. 321-325. IEEE.
- Tang, J., Deng, C., & Huang, G.-B. (2016). Extreme Learning Machine for Multilayer Perceptron. *IEEE transactions on neural networks and learning systems*, *27(4)*, pp 809-821.
- Tavallae, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). *A Detailed Analysis of the KDD CUP 99 Data Set*. Symposium on Computational Intelligence for Security and Defense Applications, pp. 1-6. IEEE.
- Thaseen, I. S., & Kumar, C. A. (2014). *Intrusion Detection Model using Fusion of PCA and Optimized SVM*. International Conference on Contemporary Computing and Informatics, pp. 879-884. IEEE.
- Usmani, S., Rehman, F., Umair, S., & Khan, S. A. (2018). A Review of Security Challenges in Cloud Storage of Big Data. *Handbook of Research on Big Data Storage and Visualization Techniques*, pp. 175-195. IGI Global.
- Zlomislić, V., Fertalj, K., & Sruck, V. (2017). Denial of Service Attacks, Defences and Research Challenges. *Cluster Computing*, *20(1)*, pp. 661-671.

# APPENDIX A



## APPENDIX B

### Evaluation of ALO-MLP classifier against most common classifiers

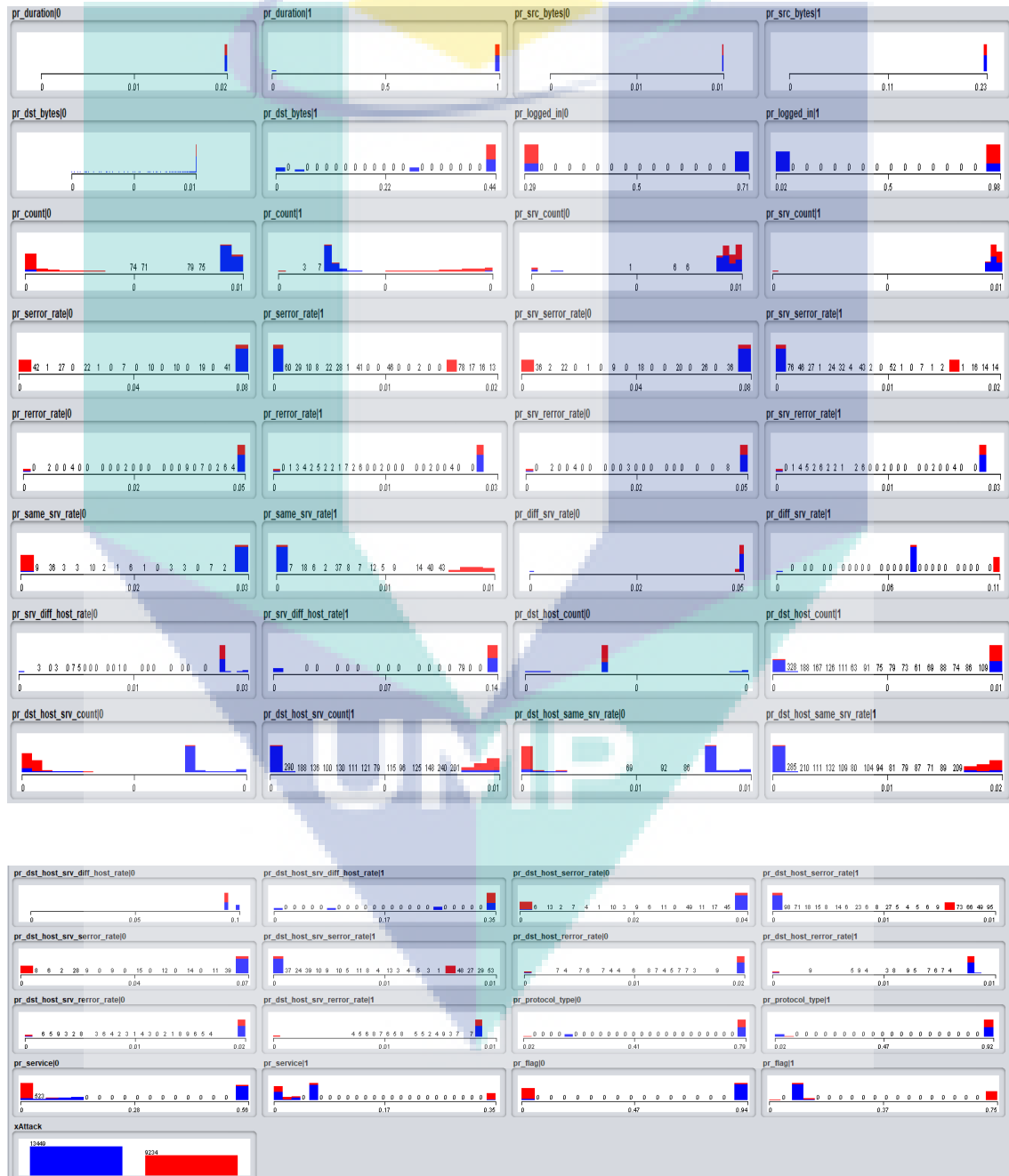
Classifier type	DoS (%)	Probing (%)	U2R (%)	R2L (%)
<b>ALO-MLP</b>	<b>99.10</b>	<b>98.80</b>	<b>98.46</b>	<b>97.37</b>
MLP	98.29	97.86	98.62	93.10
Naive Bayes	63.00	84.14	83.68	79.522
K-Means	78.00	88.00	92.00	95.00
Nearest Cluster	96.88	77.36	91.92	92.92
SVM	96.36	88.85	90.31	94.19
Decision Tree	96.35	92.07	91.41	97.51
NB tree	76.23	92.20	92.01	57.19



UMP

## APPENDIX C

Distribution histograms of all features in the original NSL-KDD training data. The x-axis shows the value of the feature and the y-axis shows how often the value exists in the training data. The highlighted features ‘num outbound cmds’ and ‘is host login’ show no variance. The highlighted features ‘duration’, ‘src bytes’ and ‘dst bytes’ have strongly biased distributions with some huge outliers



## Example of DoS in Weka

Time taken to build model: 33.55 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.05 seconds

=== Summary ===

Correctly Classified Instances	7682	99.611 %
Incorrectly Classified Instances	30	0.389 %
Kappa statistic	0.9919	
Mean absolute error	0.0107	
Root mean squared error	0.0579	
Relative absolute error	2.2265 %	
Root relative squared error	11.8108 %	
Total Number of Instances	7712	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.997	0.005	0.996	0.997	0.997	0.992	1.000	1.000	0
	0.995	0.003	0.996	0.995	0.995	0.992	1.000	1.000	1
Weighted Avg.	0.996	0.004	0.996	0.996	0.996	0.992	1.000	1.000	

=== Confusion Matrix ===

```
a  b  <-- classified as
4594 13 | a = 0
17 3088 | b = 1
```

UMP

## Example of Prop in Weka

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

```
Correctly Classified Instances      4542          99.7365 %  
Incorrectly Classified Instances    12           0.2635 %  
Kappa statistic                    0.9952  
Mean absolute error                 0.0033  
Root mean squared error             0.0393  
Relative absolute error              0.9162 %  
Root relative squared error         9.2171 %  
Total Number of Instances          4554
```

```
=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.996	0.001	0.999	0.996	0.998	0.995	0.999	1.000	1
	0.997	0.000	1.000	0.997	0.998	0.998	1.000	1.000	2
	0.999	0.003	0.995	0.999	0.997	0.995	0.999	0.996	3
Weighted Avg.	0.997	0.002	0.997	0.997	0.997	0.995	0.999	0.998	

```
=== Confusion Matrix ===
```

```
 a   b   c  <-- classified as  
2527  0   9 |  a = 1  
1   307  0 |  b = 2  
2    0 1708 |  c = 3
```

UMP



## Example of U2R in Weka

== Summary ==

Correctly Classified Instances	4777	97.3706 %
Incorrectly Classified Instances	129	2.6294 %
Kappa statistic	0.9467	
Mean absolute error	0.0438	
Root mean squared error	0.1322	
Relative absolute error	8.8902 %	
Root relative squared error	26.6499 %	
Total Number of Instances	4906	

== Detailed Accuracy By Class ==

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0	0.979	0.030	0.962	0.979	0.970	0.947	0.998	0.998	0
1	0.970	0.021	0.983	0.970	0.976	0.947	0.998	0.998	1
Weighted Avg.	0.974	0.025	0.974	0.974	0.974	0.947	0.998	0.998	

== Confusion Matrix ==

a	b	<-- classified as
2106	46	a = 0
83	2671	b = 1

UMP

## APPENDIX D

### LIST OF PUBLICATIONS AND AWARDS

1. Jaber, A. N., Zolkipli, M. F., Shakir, H. A., & Jassim, M. R. (2017). *Host Based Intrusion Detection and Prevention Model Against DDoS Attack in Cloud Computing*. International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (pp. 241-252). Springer, Cham.
2. Jaber, A. N., Zolkipli, M. F., Majid, M. A., & Anwar, S. (2017). Methods for Preventing Distributed Denial of Service Attacks in Cloud Computing. *Advanced Science Letters*, 23(6), 5282-5285.
3. Jaber, A.N., Zolkipli, M. F., Anwar, S., & Al-Hawawreh, M. S. (2016). *Present Status and Challenges in Cloud Monitoring Framework: A Survey*. European Intelligence and Security Informatics Conference. IEEE.
4. Jaber, A. N., Zolkipli, M. F. B., & Majid, M. B. A. (2015). Security Everywhere Cloud: An Intensive Review of DoS and DDoS Attacks in Cloud Computing. *Journal of Advanced & Applied Sciences (JAAS)*, 3(5), 152-158.
5. Jaber, A. N., Mohamad Fadli, Z. (2015). Security Scheme for Protecting Cloud Computing Services Against Bursty DDoS Attacks. *International Journal on Advances in Information Sciences and Service Sciences*, 7(1), 39-45.
6. Jaber, A. N., Majid, M. A., Zolkipli, M. F., & Anwar, S. (2014). *Trusting cloud computing for personal files*. International Conference on Information and Communication Technology Convergence (pp. 488-489). IEEE.
7. Jaber, A. N., Majid, M. B. A., Zolkipli, M. F. (2014). *A study in data security in cloud computing*. International Conference on Computer, Communications, and Control Technology (pp. 367-371). IEEE.

## AWARDS

1. Jun 2016 Merit awards INPEX 2017, Hypervisor IDPS: DDoS Prevention Tool for Cloud Computing.
2. Feb 2017 Gold Award at Malaysia Technology Expo for real time IDPS server design.
3. Mar 2016 Silver Medal at Creation, Innovation, Technology & Research Exposition (CITREx)
4. Apr 2015 Award: UMP Three Minute Thesis Competition.
5. Mar 2015 Award: Silver Medal at Creation, Innovation, Technology & Research Exposition (CITREx).
6. May 2014 Grant: UMP Postgraduate Research Grants Scheme (PGRS).
7. Mar 2014 Award: Bronze Medal at Creation, Innovation, Technology & Research Exposition (CITREx ).

The logo for Universiti Malaysia Perlis (UMP) is a large, stylized 'V' shape. The left side of the 'V' is light blue, the right side is light purple, and the bottom point is a darker blue. The letters 'UMP' are written in white, bold, sans-serif font across the bottom of the 'V'.

UMP