MODEL FOR IMPROVING PERFORMANCE OF NETWORK INTRUSION DETECTION BASED ON MACHINE LEARNING TECHNIQUES

JOSEPH MBUGUA CHAHIRA

A Thesis Report Presented to the Institute of Postgraduate Studies of Kabarak University in Partial Fulfillment of the Requirements for the Award of the Doctor of Philosophy in Information Technology.

KABARAK UNIVERSITY

NOVEMBER, 2019

DECLARATION

This research project is my own work and to the best of my knowledge it has not been presented for the award of a degree in any university or college.

Signature: ……………………………………. Date: ………………………………………

Joseph Mbugua Chahira

GDI/M/1151/09/13

RECOMMENDATION

To the Institute of Postgraduate Studies:

The research thesis entitled "**Model for Improving Performance of Network Intrusion Detection System Based on Hybrid Machine Learning Techniques**" and written by **Joseph Mbugua Chahira** is presented to the Institute of Postgraduate Studies of Kabarak University. We have reviewed the research proposal and recommend it be accepted in partial fulfillment of the requirement for award of the Doctor of Philosophy in Information Technology.

Signature……………………………………… Date……………………………..…

Dr. Joseph Siror.

Department of Computer Science

Kabarak University

Signature……………………………………… Date……………………………..…

Dr. Moses Thiga

Department of Computer Science

Kabarak University

ACKNOWLEDGEMENT

I appreciate all the people who have assisted and motivated me during the research and study period. I personally want to register special thanks to my supervisors. Dr. Joseph Siror and Dr Moses Thiga for the support and guidance they accorded me that helped me finish up the research and thesis writing. Their perpetual energy and enthusiasm motivated me to complete this study, their sincere concern and assistance throughout my study period are not in vain. I would also like to thank Mr Peter Kemei and all my classmates for their invaluable comments and ideas during my proposal at the early stage of my study. Special thanks goes to the respected evaluators and chairman: Prof. Simon Karume, Dr. Betty Tikoko, Dr. Maghanga,. Dr. Nelson Masese and Dr. Peter Rugiri who have given me constructive comments during defense of my proposal and thesis. My deepest gratitude goes to my dearest family and friends for their undying support and endless love. They are my true inspiration in achieving all my dreams. Last but not least, thanks to School of Computing, Kabarak University and Ministry of Higher Education (MoHE) for giving me the opportunity to further my doctorate. I wish all the best to all and may Almighty God bless them always.

COPYRIGHT

ABSTRACT

Digital crimes have increased in number and sophistication affecting the networks quality of services parameters like confidentiality, integrity and availability of resources. Network Intrusion Detection Systems (NIDS) are deployed to optimize detection and provide comprehensive view of intrusion activities. However, NIDSs generates large volumes of alerts mixed with false positives, and repeated warnings for the same attack, or alert notifications from erroneous activity. This prevents Security Analyst in evaluating the severity of each attack and selecting suitable response plan to prevent information and resources' loss in the network at the right time. To achieve high accuracy while lowering false alarm rates there are major challenges in designing an intrusion detection system. To address this issue, this work proposes a three-level model for network Intrusion detection that offers multiple types of correlations. In the first level, several feature selection techniques are integrated to find the best set of features used in this work. The existing feature selection techniques includes Correlation Feature Selection (CFS) based evaluator with Best-first searching method, Information Gain (IG) based Attributes Evaluator with ranker searching method, and Chi square and ranker searching method. The second level enhances the structural based alert correlation model to improve the quality of alerts and detection capability by grouping alerts with common attributes based on unsupervised learning techniques. This work compares four unsupervised learning algorithms namely Self-organizing maps (SOM), K-means, Expectation and Maximization (EM) and Fuzzy C-means (FCM) to select the best cluster algorithm based on Clustering Accuracy Rate (CAR), Clustering Error (CE) and processing time. Then an anomaly classification module is designed in the third level based on fusion of five heterogeneous classifiers Support Vector Machine (SVM), Instance based Learners (IBL), Random Forest, J48, and Bayes Net using Voting as a Multi-Classifier.

Network Intrusion Detection model based on hybridizing machine learning techniques (feature selection, enhanced structural and enhanced causal) is implemented on WEKA platform. This research is executed through a series of experiments and testing to achieve the goal of the research. The controlled experiment is preferred as the main method due to certain characteristics, such as performance measures, dataset evaluations and the usability of the results. The NSL KDD and UNSW-NB15 dataset are evaluated based on five measures, detection accuracy, False Positive Rate (FPR), Precision, Total Accuracy (TA), and F–Measures (FM). The results of the proposed model are compared with recent alert correlation models. The overall detection rate is 99.9%, false error rate 0.1% and execution rate of 1340.7 seconds. This shows that HAC is effective and practical in providing complete correlation even on high dimensionality, large scaled and low-quality dataset used in intrusion detection system.

Keywords: alert correlation, machine learning, model, performance, intrusion detection.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ANN -	Artificial Neural Networks
Acc -	Accuracy
AC	Alert Correlation
CAC	Causal Alert Correlation
AI -	Artificial Intelligent
AR	Accuracy Rate
Bayes Net	Bayesian Networks
c	Correlation coefficients
CA	Clustering Accuracy
CA	Clustering Accuracy
CR	Clustering Error
DAG	Directed Acyclic Graph
DOS	Denial of Service attacks
DMZ	Demilitarized Zone
DT	Decision tree
EM	Expectation Maximization
ER	Error Rate
FM	F-measure
FCM -	Fuzzy C-means
FPR	False Positive Rate
FNR	False Negative Rate
HAC	Hybrid-based Alert Correlation
HIDS	Host-based Intrusion Detection System
HIPS	Host-based Intrusion Prevention System
ICMP	internet Control Messaging Protocol
MSE	Mean Squared Error
NIDS	Network Intrusion Detection Systems
NIPS	Network-based Intrusion Prevention System
P -	Precision
PC	Principle Component
PCA	Principal Component Analysis
R’	Recall

ROC	Receiver Operating Characteristics
SA	Security Analyst
SAC	Structured Alert Correlation
SC	Statistical Correlation
SD	Standard Deviation
SOM	Self-Organizing Maps
TA	Total Accuracy
TCP	Transmission Control Protocol
TN	True negative
TP	True positive
UDP	User Datagram Protocol
U2R	User to Root
R2L	Remote to Local

DEFINITION OF TERMS

Alert Correlation (AC): is a process that contains multiple components with the purpose of analyzing alerts and providing high-level insight on the security state of the network under surveillance

Anomaly detection: analyzes a set of characteristics of the system and compares their behavior with a set of expected values. It reports when the computed statistics do not match the expected measurements

Attack is a combination of actions performed by a malicious adversary to violate the security policy of a target computer system or a network domain

Attack graph is a relational/causal graph or Directed Acyclic Graph (DAG) that represents the causal relationship between attacks to reveal attack strategy. Edges represent action and nodes represent system's state

Attack prediction process is the sequence of elementary actions that should be performed in order to recognize the attack strategy.

Attack strategy stages. A complete attack launched by attacker which consists of attack steps and attack

Causal-based AC. The correlation is emphasized on recognizing which alerts cause an attack stage for a multi-stages network attack

DDoS: stand for Distributed Denial of Service. It referred to an attack which a multitude of compromised systems attack a single target, thereby causing denial of service for users of the targeted system. The flood of incoming messages to the target system essentially forces it to shut down, thereby denying service to the system to legitimate users.

False positive – an alert that is not supposed to be reported by NIDS, typically because of flawed traffic modeling or weak rules/signatures/anomalies specified.

Intrusion detection is the process of identifying malicious behavior that targets a network and its resources.

Machine learning is a branch of artificial intelligence that acquires knowledge from training data base on known facts and allows computers to learn knowledge without being programmed.

Malicious behavior is a system or individual action which tries to use or access to computer system without authorization and the privilege excess of those who have legitimate access to the system, making it easier to understand.

Model is a representation of a particular phenomenon in the world using something else to represent it.

Performance The accomplishment of a given task measured against preset known standards of accuracy, completeness, cost, and speed.

Statistical-based AC: Works under this category correlate alerts based on statistical model to discover the relationships statistically. Alerts are correlated based on similarity of attributes such as Source IP address, source port number, destination IP address, destination port number, and time stamps

CHAPTER ONE
INTRODUCTION

1.1 Introduction

The advancement of modern computers, network and internet has led to their widespread adoption and application in organizations' critical systems. These organizations are susceptible to intrusions and malicious activities that compromises the integrity of business information (integrity and confidentiality) the loss of critical business data assets and disruption of services (availability) of system resources (Panda, Abraham, & Patra, 2015).

1.2 Background of the Study

There are different defense measures employed by most organizations to prevent the computer networks and sensitive data from intrusion or attacks like virtual private networks, authentication mechanisms, and encryption techniques. However, one of the main challenges in the security management of large-scale high-speed networks is the detection of suspicious anomalies in network traffic patterns. This is because the threats are increasing in numbers and becoming sophisticated hence it's harder to detect and find out if network traffic is legitimate or malicious due to Distributed Denial of Service (DDoS) attacks or worm propagation which exploits the weakness in an application and cause enormous security threat (Solanki, 2014).

Intrusion detection is a device that detects intrusions and forms the major defensive mechanism in a network environment (Dhanabal & Shantharajah, 2015; Thaseen & Kumar, 2013). It's main goal is to automatically monitor network traffic and classify them as normal or suspicious activities and inform the security analyst or response system to take appropriate action before the intrusion compromises the network. Generally, the IDS can be a Host-based or Network-based system depending on its monitoring capability. The HIDS operates on information gathered from within an individual computer system with regard to the internal activities and status such as system logs, application logs and audit trails (Biswas, Tammi, & Chakraborty, 2016; Subbulakshmi, Mathew, & Shalinie, 2010). Unfortunately, this approach implies a performance impact on every monitored system. NIDS performs logging of packets, immediate traffic analysis of IP networks and discovery of intrusion over the network. The advantages of using network-based intrusion detection systems are; no processing impact on the monitored hosts, the ability to observe network-level events, and to monitor an entire segment at once (Shen, Yu & Zhu, 2015). However, as the complexity and capacity of networks increases, the performance requirements for probes can become prohibitive.

Further, Intrusion detection system includes both misuse detection and anomaly detection techniques. Misuse IDS deals with the detection of intrusion by comparing their parameters with its pre-learned scenario in the network system. However, Misuse detection Pundir & Amria, (2013) requires a learning algorithm to be trained by a dataset in which each instance is identified as a normal class or an anomaly. This algorithm cannot identify novel attacks not included in the training set but can learn the new attacks through a new training dataset. Misuse detection IDS must continuously be updated with rule-sets and upgrades to be up-to-date with the recent threat vectors. The biggest companies like Norton releases new rule-sets on a regular basis, but even these might not be sufficient. Anomaly is the process of constructing models of normal network events and identifying the events that deviate from these models detection (Dhanabal & Shantharajah, 2015; Jain & Rana, 2016). Anomaly detection systems can identify attacks if at all the attack behavior drastically changed from the normal system behavior or network profiles by making profiles of normal network scenarios or by using the system behaviors. Anomaly detection suffers from high false alarm rate as many unobserved normal events are also considered as anomalies (Mallissery, Kolekar, & Ganiga, 2014).

Network monitoring systems are used by security experts to investigate the health status of organization network and conduct further research in alert correlation, forensics investigations and anomaly detection. Due to changes in technologies, several obstacles have emerged that reduces the effectiveness of NIDSs. Firstly, they have generated huge volume of low quality evidence and in different format produced by distributed IDS systems (Application, Network and Host based). The alerts generated are either false positive (benign traffic that has been classified as intrusions) or false negative attacks (attacks that are not successful). This has increased the training and testing correlation processes, reduced effeciveness of the system, reduced the detection accuracy and increased performance costs. The preprocessing and normalisation phase should be introduced in the system for filtering and labeling records before analyses (Wahba, at el, 2015). Secondly, the continuous development of new and sophisticated attacks strategies like hidden attacks, coordinated, slow and low attacks which use stealth and intelligence to strategically compromise a target, escaping detection and penetrating the defenses (Amini, 2014). Lastly, IDS alerts does not contain sufficient information on which security administrator can take decision as they work in limited domain and in single model (Siraj, Elshoush, & Elhaj, 2016). As a result, the systems cannot differentiate between normal and abnormal traffic with high level of accuracy. Thus, researchers need to develop more reliable, effective, and self-monitoring systems that can adapt to the continuous changes occurring in modern networks. The system, undergoing such kind of attempts, catastrophic failures of susceptible systems can be reduced.

Alert Correlation (AC) takes the generated alerts, process and produce compact reports on the security status of the network under surveillance (Chakir, Moughit, & Khamlichi, 2017; Sendi, Dagenais, Jabbarifar, & Couture, 2012). Alert Correlation is a multiprocessing component aimed at improving the performance of IDS by removing redundant alerts, extract attack Strategies and predict attacker's next course of action. AC improves the quality of alerts through preprocess alerts to eliminate redundancy and irrelevant alerts based on feature selection. The system develops attack graph based on the correlated alerts in the detection and prediction component. This component assists the Security Analyst or Intrusion Prevention Systems (IPS) to react appropriately before the network is compromised. Performance measures of IDS includes True positive rate (TP), the classification rate and time taken to build the model (Chaudhari & Parikh, 2012). However, the research in this area focuses on improving the performance of IDS based on these metrics.

There are four main techniques proposed in alert correlation focusing on analyzing intrusion alerts produced by computer networks to improve detection and prediction ability in NIDS. In Structural-based AC (SAC), the correlation of alerts is based on comparison of features in the alert. The Similarity index or function is computed to determine the degree of relationships between alerts components. SAC is capable to ascertain known group of alerts and the attack steps, research by Siraj, Maarof, & Hashim, (2009), confirmed that SAC cannot determine the cause effect relationships among alerts. The Causal-based AC (CAC) analysis finds the relationship between alert types in the alert stream to discover alert attributes that have the greatest impact on the relationship between intrusion alerts. Research by Siraj, Hussein, Albasheer, & Din, (2015); Govindarajan, (2014); Song, (2016) have showed that the technique can discover unknown alerts but building a comprehensive attack database for every attack action with its pre- and post-conditions is very expensive. The Statistical-based AC (STAC) defines normal behavior by collecting data relating to the behavior of legitimate users over a period of time. The work by Shameli Sendi, (2013); Thomas & Balakrishnan, (2008) indicates that optimum results from Statistical-based AC largely depends on how the parameters are set which is hard to achieve. The goal of data mining and machine learning technique is to built a model expressed as an executable code and can perform feature selection, clustering, classification, prediction or other data mining activities. This approach employs anomaly detection techniques and does not require prior knowledge about attack scenarios to identify new attack scenarios. The approach is time consuming during training and testing (S. Kumar & Naveen, 2016).

Machine learning is a branch of artificial intelligence that acquires knowledge from training data base on known facts (Panda, Abraham, & Patra, 2012; Parsaei, Rostami, & Javidan, 2016). It allows computers to learn knowledge without being programmed. Machine learning techniques are incorporated to build IDS capable of modeling intelligent decision with high detection rate and low false alarm rate. Feature selection (Techniques selects accurate and significant features from intrusion dataset to have better results and less computational time Biswas et al., 2016). Clustering is a machine learning techniques applied with unlabeled data set and can detect unrecognised attacks in large datasets (Kansra & Chadha, 2016). Classification is used to identify the class of label of instances based on the attributes in a dataset (Balakrishnan, K, & a, 2014). Scholars have tested different classifier models to solve intrusion detection problem, such as rule-based detection Palanisamy, (2006), Neural Networks (Haddadi, Khanchi, Shetabi, & Derhami, 2010; Kidmose, Stevanovic, & Pedersen, 2016; Kuźniar & Zając, 2015), fuzzy logic (Mukosera, Mpofu, & Masaiti, 2014; Sendi, Jabbarifar, Shajari, & Dagenais, 2010), random forest model (Hasan, Nasser, Ahmad, & Molla, 2016; Pundir & Amrita, 2013) and Bayesian analysis (Barber, 2010; Bouckaert, 2008; Kohavi & Mateo, 1999).

Several machine learning methods like Decision Tree, Naïve Bayes and Neural networks perform well for the detection of attacks having high frequency data in the dataset. But some renowned datasets like NSL-KDD Assi & Sadiq, (2017); Ingre & Yadav, (2015) contain some low frequency attack classes like Remote to Local (R2L) and User to Root (U2R). In Long, Wang, & Zhu, (2015), the low frequency class detection rate is lower than high frequency attack classes because the training sample for low frequency classes is too small compared to high frequency classes. As a result, the detection precision degrades which is a drawback while using single classifiers in developing IDS as they cannot detect multiple class categories (Long *et al.*, 2015). The low frequency attackers may feel comfortable to use them and hence the detection rate of the IDS will be under threat. Different measures are used to check the efficiency and accuracy of an IDS system which includes prediction performance, time performance and fault tolerance (Chahar, Gigras, & Singh, 2017). Prediction function involves the classification rate of true and false classified network traffic. Systems performance depends on correct prediction rate. If rate decreases, then false positive rate will be high. Time performances is the rate at which IDS is generating and propagating the results to resolve the attacks. The other factor is fault tolerance which requires IDS to be robust and ability to recover from attacks.

1.3 Statement of the Problem

The existing work in alert correlations techniques based on structural, causal and statistical models are unable to discover complete relationship among known and unseen alerts due to low quality alerts produced by NIDS and unrecognized attack strategy. Machine learning can enhance the overall performance of intrusion detection systems by reducing the feature set, cluster unlabeled data in large data set to detect known and unseen attacks and classify the anomalous traffic into their attack types. Recent research on Intrusion Detection Systems (IDSs) based on machine learning focuses more on improving the detection rates of machine learning classifiers (Alhaj *et al.*, 2016; AliShah, Sikander, & Daud Awan, 2015; Chand, Mishra & Govil, 2017; Miškovic, 2014). The findings indicate that despite the high detection accuracies being achieved, there is still room for improvement in areas such as the dependence on human operators, long training times, lower detection precision especially for low frequent attacks like Remote to Local (R2L), User to Root (U2R). It can determine the relationships between known and new alerts produced by multiple NIDSs comprehensively. (Shone, & Shi, 2018). Reliance with the current detection techniques will result in ineffective and inaccurate decision from the Network Security Analyst (NSA).

1.4 Purpose of the Study

The broad objective of the study was: To design alert correlation model for Improving performance of network intrusion detection based on hybrid machine learning techniques

1.4.1 The specific objectives :

i. To determine the optimum features based on hybrid feature selection techniques
ii. To enhance the structural based alert correlation model using unsupervised machine learning techniques.
iii. To enhance the causal-based alert correlation model using supervised machine learning techniques.
iv. To design an alert correlation model based on hybrid machine learning techniques to enhance the performance of network Intrusion Detection Systems.
v. To validate the model based on derived metrics and comparisons with current alert correlation models

1.5 Research questions

i. What are the relevant feature set based on feature selection techniques to improve the classification accuracy and to have less resource consumption?
ii. To what extent can the structural based alert correlation model be enhanced using unsupervised machine learning technique in order to improve the quality of alerts and identify attack steps?
iii. To what extent can the causal-based alert correlation model be enhanced using supervised machine learning techniques in order to discover the relationships among alerts based on their causes?
iv. What is a suitable hybrid-based alert correlation model for detecting and predicting intrusions in computer network?

v. How does the model be evaluated based on derived metrics perform compared to current alert correlation models?

1.6 Significance of the Study

Alerts generated by multiple NIDSs are meaningless unless they are analyzed through correlations. The knowledge gained from this work will enable the Security Administrator to investigate, design and develop an accurate and appropriate alert correlation system. Analyzing alerts generated by intrusion detection systems is a challenging task as a result of the huge amount of low quality alerts generated by NIDSs (Thaseen & Kumar, 2017). The feature selection techniques will help obtain a feature set that is comprehensive enough to separate normal data from intrusion data and also keep the size of this set as small as possible. The enhanced structural alert correlation model using Expectation Maximization (EM) unsupervised machine learning techniques aims to improve the quality of alerts and reveal the list of attack steps by aggregating similar alerts as well as filter the low quality alerts. Principal Component Analysis (PCA) is implemented to reduce the alerts dimensionality and optimize the performance. This technique is able to facilitate improved classification results.

The enhanced causal alert correlation model based on machine learning (Staking SVM with five other machine learning algorithm namely Random Forest, C4.5, Bayesian networks and Artificial Neural Networks) will help to recognise the memberships of several attack stages of a network attack and to improve the performance of anomaly detection system based on high detection rate, high execution speed and low false positive rates. The proposed hybrid alert correlation model hybridizes the feature selection techniques (chi square, Information Gain and Correlation based), enhanced structural (Expectation Maximization) and enhanced causal alert correlation models to optimize the performance of the overall network intrusion detection system. Unlike other systems which requires updating rules frequently to discover attack strategy which are less practical and required high costs due to large database and labour intensive. It is capable of facilitating a deeper analysis of network data and faster identification of any anomalies.

1.7 Limitation of the study

This work analyses the NSL-KDD datasets widely used benchmarks and hence it exhibits some weaknesses like other research works which use publicly available benchmark data enabling to draw direct comparisons.

This research focuses on analyzing the alerts to predict intrusion strategies as a guide in designing Intrusion Responsive System (IRS). IRS are implemented in a network to provide an effective response mechanism. Hence, design of the responsive system is excluded in this work.

The improvement on quality of alerts is based on elimination of false positive, invalid and redundant alerts. Several other limitations of current IDS in alert correlation (AC) such as prediction and real time response are not addressed. The identification of network traffic as normal or abnormal and classification of the attack type as probe, User to Root attack, Denial of Service, Remote to local attacks. The NSL KDD dataset is evaluated based on five measures, detection accuracy (or) True Positive Rate (TTR), False Positive Rate (FPR), Precision, Total Accuracy (TA), and F–Measures (FM).

1.8 Conclusion

NIDS generate huge volume of low quality evidence and in different format produced by distributed IDS systems. To solve these problems, this work proposed a novel approach based on hybrid machine learning techniques in order to improve the detection stability and detection precision in NIIDS. The system will aid the Security Analysts to automatically analyze intrusions and make optimal decision before the organization networks are compromised. An empirical investigation to improve the detection ability of Intrusion detection system was conducted and several methodological factors, such as choice of data subset based on feature selection, dimensional reduction and combination of classification methods were considered.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter presents an overview of Intrusion Detection Systems (IDS), the various categories of IDS, the challenges in IDS and problems in alert correlation (AC) techniques. The comparison and limitations of existing models are also discussed. Based on these restrictions and advantages of hybrid machine learning techniques in handling AC problems, the conceptual framework for the study was developed.

2.2 Intrusion Detection Systems

Intrusion is a combination of events that constitutes a security incident in which an intruder gains access to a system without having authorisation to do so. The attackers use sophisticated means to exploit vulnerable systems to gain remote access to computer host over the network, gain unauthorised additional user privelleges or misuse privelleges granted for malicious gain (Verma, 2016). An intrusion detection system (IDS) comprises of either hardware or software that facilitates network's resistance against external attacks. The IDS inputs is information from several sources within the computer system and from the organisation network and then matches the collected information against its database of attack signatures or normal behavior profiles, identifies potential attack scenarios, and responds to events with signs of possible incidents of violations of security policies (Albayati & Issac, 2015). They are able to identify the intruders both from inside and outside as well as within the organization and also provide important information against an intruder in a timely manner to prevent further occurrence or escalation. Intrusion detection systems generate enormous amount of alerts which are of low quality and hence, the attacker execution plans cannot be discovered directly from the alerts. Hence, human intervention is required to investigate the attack once it is detected and reported.

1.8.1 2.2.1 Categories of Intrusion Detection Systems

Intrusion detection system can be divided into two: Host Based Intrusion Detection System (HIDS) and Network Based Intrusion Detection System (NIDS) based on location. Host Based Intrusion Detection System (HIDS) consist of software or agent components and can run on the server, router and switch or network appliance (Gervais, Munif, & Ahmad, 2016; Mahmood & Hussein, 2013). Since HIDS monitors the host system, they can be detected by attackers and can get tampered with or even shut down. They can provide information source such as system commands, system accounting information, process logging information and security audit information. NIDS continuously monitor information from the system audit data or network activities logs and automatically generates and sends a warning in case they detect a security violation. The Network Based Intrusion Detection System (NIDS) Pathak, Sharma, & Srikanth, (2010) collects network traffic packets such as Transmission Control Protocol (TCP) and User Datagram protocol (UDP and analyzes the packet pay loads looking for dangerous contents. HIDS and NIDS are needed to complement each other as they function at different levels in a computer system. Passive systems are called Intrusion Detection Systems and reactive systems are referred as intrusion Prevention Systems. IDS notice malicious activities from a set of log histories and alert the user. IPS reply by reprogramming the firewall to block network traffic from the suspected source.

IDS can also be grouped as anomaly detection and misuse based on detection mechanism. The misuse detection systems Kang, (2015), Noureldien & Yousif, (2016) utilize a rule database that explicitly models what is not allowed. Everything that does not match any of the rules is allowed. Misuse detection systems are further divided into stateless and stateful systems (Fanfara, Dufala, & Radušovský, 2013) . A stateless IDS analyses the existing audit events when determining whether an event is good or malicious. In contrast, stateful systems stores and analyses the information from previous audit event and if an incident is reported, both the recent event and the sensors state are taken into account in the detection process. The stateful systems can support more complex rules than stateless systems. The stateful sensors consume memory in order to store the state. Stateful sensors requires more processing power than a stateless system as the rules are more complex.

Anomaly detectors Hajamydeen, Udzir, Mahmod, & Abdul Ghani, (2016) can be either learning-based or specification-based. In anomaly detection, network behavior is represented as normal or valid behavior and later used to compare the current network activity. When a deviation arises an alarm is generated indicating an intrusion. This approach that generates the log records of IDS contains Panda *et al*., (2012); Science, (2015) large number of features which make the task of an Intrusion Detection System very hard. Hence, significant features can be achieved by the use of reduction algorithm to classify data either normal or attack. They produce high false-alarm since the entire performance of an information system may not be enclosed during the learning phase and also behavior can change with time. The misappropriation of intrusion-detection technique applies the knowledge accumulated about specific attacks. An alarm is raised when an attempt is detected. Specification-based systems depend on the specification of how the normal traffic resembles. This specification may be specified physically or computerised produced. Computer generated specification can be supportive, because it is very hard to physically stipulate how the normal events look like. computer generated can also yield specification without errors, while a physical specification produce a lot of errors. Devices that automatically create specification that, for example, been applied by schemes checking for illegitimate arrangements of system calls. A profile of the acceptable system call classifications can be computerised mined at the source code of the secured program. When revealing, the system call arrangements produced by the program are matched with the profile of acceptable sequences.

The misuse detector has higher accuracy when compared to anomaly detector because representing normal network behavior is difficult and allows an efficient implementation (Sunita, Chandrakanta, & Chinmayee, 2016; Valeur, 2006). The major limitation of this technique is the frequent updates needed to keep up with new threats discovered and it is not possible to keep all malicious signature actions on the database (Baykara & Das, 2015).

Intrusion detection systems can as well be classified based on their usage frequency. The online IDS function in real-time and utilises audit data as is produced. This is no time lag between when the information is generated and processed. The other systems operate in offline mode; in this case the system run occasionally to check for signs an intrusion. Intrusion detection systems can also be classified based on the type of reaction the system implements whenever an intrusion is discovered. Most common kind of reaction is passive where the intrusion incident is recorded or the security analyst is notified by other means (example., SMS or email). Active systems block an attack so it cannot succeed. These systems are usually referred to as intrusion prevention systems (IPS). Depending on the implementation, an active system could, for instance, send a reset packet to tear down the attacker's connection or update the firewall rules so that the attacker is blocked.

When the alert correlation component is dependable, the subsequent phase would be to safeguard the network based on protective activities by means of Intrusion Prevention System (IPS). Intrusion prevention system is an access control system that drops discovered bad traffic flow in real-time by blocking the flow to its end point. It is applicable in detecting denial of services floods, brute force attacks, vulnerability detection, protocols anomaly detection and protection against nameless attacks (Stakhanova et al., 2007). Intrusion prevention system can be attained with three main approaches: 1) Constructing devices with less susceptibility, 2) Taking faultless corrective steps to determine weak areas and reinforce them, and 3) Discovering the exploit attempts and blocking them before serious damage is done.

There are four different types of methods in the Intrusion prevention system to protect the network (Sendi et al., 2012): 1) misuse -based. patterns are included in the system that detect a pattern with the most common occurrences existing. These patterns can be added, adjusted, and regular updating to handle the novel attacks. 2) Anomaly-based. It endeavors to identify traffic that deviates from what the administrator regards as standard action. 3) Policy-based. Alarms are generated when activities are discovered that violates the security rules implemented in the organization and inscribed into the Intrusion prevention system. 4) Protocol Analysis-based. It is related to misuse-detection technique. however, this technique performs in deeper packet analysis and is more efficient in detecting specific attacks types.

1.8.2 2.2.2 Challenges in Network Intrusion Detection Systems

Network Intrusion Detection Systems has been used extensively for the purposes of security, forensics and anomaly detection. However, recent advances have created many new obstacles for NIDSs. Some of the most pertinent issues include:

i. Low performance of NIDSs

The traffic in the network, whether stored or in motion has drastically increased. To operate at such speed and ensure satisfactory levels of accuracy, effectiveness and efficiency presents a significant challenge in NIDS. Consequently, huge volumes of alerts are generated which are not caused by real attacks (Ben Mustapha, 2015; Gorton, 2003; Manandhar, 2014; Shameli-Sendi, Desfossez, Dagenais, & Jabbarifar, 2013). These alerts are called false positives and this is reason why in several instances NIDSs are used together with a human expert or security analyst. True positives warnings are very significant in defining correct reaction and defensive actions

ii. Missing Information

The contents of intrusion reports produced by NIDS contains less information to support decision making by the administrator (Valeur, 2006). Nowadays, the number of new or customized protocols being implemented in modern networks have drastically increased. As a result, distinguishing between normal and abnormal traffics with high accuracy is difficult (Shameli Sendi, 2013). A network-based sensor only identifies network-based attributes like

IP addresses and port numbers. It provides incomplete report of the security status of the device it is protecting since it does not take into account either the process id or user id of the processes that are accepting the network connections it observes. Likewise, host-based sensors generate incomplete reports about the behaviors of an attack. A host-based sensor that reports a buffer overflow in a program normally does not include the IP address of the attacker because the sensor doesn't have access to such information. A meaningful prioritization score is similarly missing from the alerts of most IDSs (Shameli-Sendi & Dagenais, 2014). Network-based sensors usually do not discriminate between attempted attacks and successful ones. This, in combination with the vast amount of alerts usually produced by sensors, makes it very hard to get a high-level picture of the security state of the network (Chakir *et al.*, 2017; Siraj *et al.*, 2009b). This prevent the Security Analyst to deduce the actual meaning signified by the alerts. The need to correlate alert to recognition of attack strategy to assist SA in determining the conduct of an attacker and review the attack strategy.

iii. Non-Contextual alerts

A non-contextual alert is an intrusion detection alert generated as a response to a real attack, but because of the configuration of the host, the attack cannot succeed alert (Kenaza & Aiash, 2016; Yusof, Selamat, Sahib, Mas'ud, & Abdollah, 2011). For instance, an alert warning approximately that of a web-based attack produced by Windows based operating system, while the target is a Linux box. The NIDS possess less information regarding the hosts they are protecting (Valeur, 2006). Low-frequency attacks prevent the earlier anomaly detection techniques, including artificial intelligence approaches. The problem originates from imbalances in the training dataset, meaning that NIDS offer weaker detection precision when faced with these types of low frequency attacks.

iv. Adaptability and dynamics

Current networks have employed advanced technologies to decrease their dependences on stagnant technologies and administration styles. Consequently, there is more prevalent application of dynamic equipment such as containerization, virtualization and other software implementable networks. Detection systems should adopt the utilization of such current techniques with the limitations they cause (Rahayu et al., 2010). With the variety and flexibility of current networks, the behavior is flexible and difficult to calculate. These behaviours results in difficulty in computing a reliable behavioral norm. It raises doubts in the lifespan of learning models.

v. Low-frequency attacks

The low frequency attacks have prevented the earlier anomaly detection techniques, like artificial intelligence approaches from achieving a high detection rate. The challenges originates from imbalances in the training dataset, implying that network intrusion detection systems have weak detection capability (Shone et al., 2018).

vi. Diversity

Current years have witnessed an increase in the number and frequency of new and customised protocols being employed in modern networks (Shone *et al*., 2018). The number of devices connected to the network and/or Internet connectivity contributes to this scenario. consequently, differentiating between normal and abnormal traffic with high precision is difficult.

2.3 Alert Correlation System

Alert correlation consists of several processes that accepts alerts from one or several inputs from IDS and generates an advanced report of the malicious activity on the network. To obtain acceptable results, the data should be collected from various sources to offer additional coverage of the attack vector space. In this case, the correlation engine measures the degree of the attributes from the same group of elements (clusters) which tends to vary together. Alert correlation can improve the detection level and provide a comprehensive scenario of the attacks strategy as compared to an individual sensor when varying investigative techniques supplementing each other and the resultant report have a complimentary coverage of the various logs resources (Alhaj et al., 2016).

The sources of data for correlation can be categorized into two: primary evidence and secondary evidence. The primary evidence is the reference information that triggers or indicates attacks or security policy violations and is generated by sensors designed for security investigation such as Intrusion detection systems (IDS), Firewall, vulnerable scanners and performance monitors. The secondary evidence does not directly indicate an attack or security violations but can be used to discover hidden suspicious events that evades detection by the specific security sensors but have played a role in coordinated attack. Also, they can be used to evaluate the trustworthiness of the primary evidence and generated in a much higher volume.

There exist three alert correlation techniques which includes: similarity-based, causal based and data mining technique of individual attack. To make clear the dissimilarities between them, the current works can be classified based on the principles or tactic employed to correlate them. In precise, SAC models are based on solving the problem of enhancing the quality of alerts while CAC and StAC models deals with the problem of identifying the attack approach. For each respective group, a detailed description is provided in the subsequent sub-sections together with the practical techniques.

1.8.3 2.3.1 Structural-based Alert Correlation

This approach focuses on improving the quality of alerts reduction by comparing an alert with the same attributes or features (such as Source IP address, source port number, destination IP address, destination port number, and time stamps). A correlation score is calculated between these alerts before creating an association of alerts who's the similarity degree index is high else a new thread is created if none is match depending on the score. This technique is widely used due to its simplicity to implement, though is unable to detect composite attacks due to its dependence on expert knowledge to determine the similarity degree between attack classes (Alhaj et al., 2016). furthermore, it does not discover the causal relationships between alerts when alerts of different attributes have been grouped into a single attack. Not all the attacks in this case will be detected.

The Collection mechanism and reduction of IDS alert framework (CMRAF) (Chaurasia & Jain, 2014;Chaurasia *at el*, 2014) was proposed to reduce the amount of duplicated IDS alerts and minimize the number of false alerts. They applied the information gain ratio algorithms to obtain the similarities that exist between sets of alerts and generate the maximum weight to the features based on the class of alerts belonging to the algorithm.

Alert correlation model based on novel clustering approach (Elshoush, 2014b) applied an incremental clustering approach to reduce the amount of alerts generated by IDS. Three attributes: destination IO, signature-id, and timestamp had been extracted and hashed by using MD5. The hash value from the next input topple is checked against hash value of the existing clusters. The hashing technique is used to speed up the comparison in checking the similarities of alert attributes.

An enhanced framework for intrusion alert correlation was developed by Elshoush *at el*, (2012) This research separated alert correlation framework into ten main components and categorized them as Data Normalization Unit (DMU), Filter-based Correlation Unit (FDU) and Data Reduction Unit (DRU). Similar alerts are fused based on seven extracted features namely: Event ID, timesec, SrcIPAddress, DestPort, DestIPAddress, OrigEventName, and SrcPort in order to remove duplicate alerts created by the independent detection of the same attack by different sensors.

A probabilistic-based approach proposed by Valeur (2006) correlate and aggregate security alerts by measuring and evaluating the similarities of alert attributes. They use a similarity metric to fuse alerts into meta-alerts to deliver a complex-level assessment of the security state of the system. Alert aggregation and scenario construction are conducted by enhancing or relaxing the similarity requirements in some attribute fields. But similarity correlation is the only way for them to aggregate the alerts. They have to compare all the alert pairs and have to determine lot of thresholds with expert knowledge which lead to their huge volume of computing workload.

1.8.4 2.3.2 Causal Based Correlation

Correlating alerts grounded on their causes is known as Causal-based Alert Correlation (CAC). The cause for an alert is important to be revealed in order to identify the targeted system/software vulnerabilities installed in the network. Essentially, the attacker will take benefit of these weaknesses to illicitly intrude the secure networks. These efforts can be signified as attack phases for the network attack. The correlation system comprises of three main modules including: IDS, ALERT and ATTACK. These modules are close connected and their associations are defined using the coded language as needed to connect them together. For each element, its own predefined constraints or features. The ALERT is produced by the IDS and it state that intruder has been discovered. The launched ATTACK is discovered by IDS that activates the ALERT. Meanwhile the connections should be physically coded, those model are limited to well-known attack steps. Furthermore, explanations of the attack rely on the expert's information that is labour intensive, proned to error and unreliable.

The causal correlation can be categorized into:

2.4 Predefined Attack Scenario

Predefined attack employs the fact that intrusions frequently needs numerous actions to take place in order to prosper. Each attack state has matching steps required for the effectiveness of the attack. Low- level alerts from IDS are equated with the pre-defined attack state before the alerts can be correlated. It can detect only known attack and those specified by human users or learned through training datasets. The improvement with this method is its ability to accurately detect documented attacks derivative from the libraries. In case of a novel attack, the method is unable to detect the intrusion. The main limitations of this approach, it requires more comprehensive scenario libraries and the time and cost to develop and uphold them. Comprehensive knowledge on various attack strategies is prerequisite to manually define the attack conditions. hence, it is not suitable for complex and large-scale networks (Farhan *et al*, 2012; Siraj *et al.*, 2015; Urvashi *et al* , 2015).

An alert fusion model is inspired by artificial immune system, (Farhan, *et al.*, 2012) which is an aggregating method moved by Danger Theory and combines several produced alerts grounded on the prediction of attack scenarios. The research groups the alerts into One-to-One, Many-to-One, and One-to-Many. All the alert is grouped and a priority value is computed. The grouped alerts in each group is validated with already defined rules for the possibility of raising the danger alarm by using the Danger Theory.

Automatic attack scenario discovering based on a new alert correlation method automatically mine multi-step attack scenarios was introduced by Siraj *et al*. (, 2015). The alert is associated with the alerts scenario it belongs to and then inserted in an alert tree. The scenarios create Sub scenarios and meta-alerts which are extracted to construct the multi-step attack graph for each attack scenario.

A novel alert correlation algorithm was proposed by (Balakrishnan *et al* , 2014). The researcher developed an algorithm generated on signature-based network IDS (NIDS) called pseudo Bayesian alert correlation. It is based on Bayes’ theorem of conditional probability and was aimed to recognize the groups of intrusion alerts depending on their recognition time and on the previous alert information produced by the similar system. The former information is examined occasionally while current intrusion alerts were gathered from the NIDS and examined in real time.

2.5 Prerequisites and Consequences of Attacks

Current attacks are not independent but have a relationship with each other at separate stages of attack arrangements with the previous attacks making the way for resultant attacks. For the attack to be successful, the precondition of an attack is essential (Ning *et al*, 2004). In this method, an established criterion is used to train the causal effect association amongst alerts and the weights in such relations. The advantage of Prerequisites and Consequences method is that they analyst are not required to stipulate all the possible situations but they are able to detect unknown attacks. however, it is costly to build a complete attack database with every attack action with its pre- and post-conditions (Kamesh *et al*, 2014; Siraj *et al*, 2015). Also this approach is not applicable design current networks due to their complexity of the design and user behavior.

The Bayesian network alert correlation does not require expert knowledge to discovers the attack plans Sampath *et al*. (2016). The approach use classification to extract attack plans by taking into consideration the arrangement of actions. It takes advantage of historical data generated from log and based on observed intrusion objective classifies them as class variables. The likely attack states built from hyper alerts arrangements are scrutinized and the most reasonable approaches for building accommodating attack are mined. Osman (2012) used an on-line prerequisite-consequence-based correlation method to examine and determine attack setting after alerts. The conditions that the constituent attacks are usually not lonely, but there is a relation at different stages of the attacks, with the quick ones arranging for the future ones. They introduce the notion of hyper alerts to represent the prerequisite and the consequence of each type of alert by using logical predicates. Each hyper-alert is a tuple (fact, prerequisite, consequence), where fact is the set of alerts attribute's names, and prerequisite and consequence are two different sets, each one consisting of a logical combination of predicates expressed as mathematical conditions on the variables contained in the set fact. The prototypical employs dispersed mediators to gather alert evidence online and implements prerequisite-consequence correlation technique to examines and determine attack set-up and resolved intrusion behind the alerts.

An alert correlation technique that was based on causal approach, was proposed by Kamesh *et al*. (2014). The researcher, represented the knowledge base of attack patterns as a graph model called causal relations graph. The trees connected to alerts likely correlations are constructed offline while the correlations of each received alert in real time with previously received alerts will be identified by performing a search only in the corresponding tree. Wahba, *et al*.(2015) developed a rule based correlation language MARS, a Multi-Stage Attack Recognition System which is based on prerequisite-consequence correlation method . Unlike others, they add another two parameters for modeling attack consequences like vulnerability and extensional consequences. MARS is mainly based on the phenomena of cause and effect. It has two main components: online and offline. The main purpose of the online component is to receive raw alerts and generates hyper-alerts. Then, multi-stage attack recognition is applied to correlate hyper-alerts based on rules provided by the offline component

Thomas *et al*. (2008), proposed alert correlation model based on prerequisites and consequences of individual detected alerts. A knowledge database "Hyper-alert Type Dictionary" contains rules that describe the conditions where prior behaviors prepare for later ones. Attack strategy is represented as a Directed Attack Graph (DAG) with constraints on the attack attributes considering the temporal order of the occurring alerts. The nodes of the DAG represent attacks and the edges represent causal and temporal relations. Similarities between these strategies are measured to reduce the redundancy. A technique of hypothesizing and reasoning about missing attacks by IDS is presented to predict attribute values of such attacks. The significance of their work is the reduction of the huge number of security incidents and to report a high-level view for the administrator. However, the proposed system is useful as a forensic tool where it performs offline analysis. In addition, building the knowledge database containing rules of the applied conditions is a burdensome. and authors have not provided a mechanism to build the Hyper Alert dictionary.

2.6 Statistical Models

Statistics model collects data in a profile. Analyzing the profile of normal statistical behavior gives a description shown by the patterns from data to help make conclusions if an activity is normal or abnormal. The system then develops a distance vector for the observed traffic and the profile. An alarm is raised by the system when the distance is great enough (Chaurasia & Jain, 2014; Moustafa, 2015; Mukosera *et al.*, 2014). These models are sub categorised into four: mean/standard deviation, multivariate, Markov process and operational model. Sampath *et al.*, (2016) discusses model based on the hypothesis that security violation can be detected by monitoring a system's audit records for changes in pattern. The model includes profiles for representing the behaviour of subjects with respect to objects in terms of metrics and statistical models and rules for acquiring knowledge about this behaviour from audit records and detecting anomalous behaviour. Qin and Lee (2003) applied Granger Causality Test Vidal (2017) Framework focuses on correlation of the alerts released from network intrusion detection systems centered on statistical analysis of the payload for recognition of anomalous content. The approach is driven by the need to supplement the Advanced Payload Analyser Preprocessor (APAP), which is an Anomaly Based System. The general-purpose alert correlations overlooked the nature of the sensor, which is realized by studying specific features of the payload-based detection. The framework also concentrates on association of occurrences suited to similar sensors in a multi layered system that permits their individual and group treatment.

Mustapha (2015) proposed the honeypot based alert correlation and enforcement based alert correlation that groups alerts processed by a common policy enforcement point. honeypot based alert correlation used local knowledge from IDS and Firewalls with information from global view resulting in coverage limitation, unfilled attributes and reference such as threat type, the weakness that can be exploited, lack of standard data illustration and cross reference problem. Since the enforcement alert correlation approach closely depends on the approximation of the firewalls and their configurations, some generated alerts will not be correlated.

Fredj (2015) used graph based correlation that incorporates the Context based control system that drops all alerts reported by the IDS that could not have any effects according to the saved security policies thus reducing the false positives alerts however real time false negative alert detection remains a challenge. Even with a good correlation system the researcher recommends for an automatic reaction system for future networks and applications.

Ramaki, *et al* (2015) used an alert correlation framework that mines causal knowledge based on Bayesian network. The intention of the proposed framework was to overcome the following challenges, the generation of the low level alerts. The researcher solved the problem by validation process that used vulnerability database and the topology database and tried to find a logical association amongst them. The other disadvantage is that alarms of a multi stage attack cannot extend to the correlation module simultaneously.

The third challenge was the received alerts were analyzed and omitted upon arrival because in real time applications the memory is limited. To overcome the second and third challenge the researchers used the probability propagation. Having an online and offline mode in the alert correlation framework the alerts were first collected and in the online mode the causal dependencies between the meta-alerts and probability of transition between them was calculated. Some meta-alerts were precondition for other meta-alerts to be generated. The investigator did not consider the false negative alerts.

Wang, *et al* (2016) proposed an alert correlation method which any suspicious packet found by the IDS an alert was raised and stored in the alert database. The alert filter components filtered duplicated alerts. Features between new and old alerts were extracted and the equality constrain set (relationship of the attribute values of two alerts) was calculated and stored in the equality constraint set table. Different weights based on different (Equality Constraint Set) ECSs was chosen and correlation cell value between the new alert and old alert was calculated. The correlation cell value was then added to the corresponding cell of the alert correlation matrix and attack graphs were generated based on alert correlation matrix. The attack graphs were used to model the system vulnerability and formulate attack scenarios, however the researchers focused on eradicating the alert duplication but did not consider reducing the false positives and false negatives. The other shortcoming of the proposed method was even if the vulnerabilities of the attacks were corrected, the corresponding cell values stored in (alert correlation matrix) ACM still remained large. (Benferhat & Sedki, 2012) proposed a new alert correlation approach based on knowledge and preferences of security operator. The purpose of the research was to develop a logic that represent security operator knowledge and preference and then develop an inference mechanism that ranked and ordered alerts and classified them in groups that the security administrator can analyse manually. The alarms from the first group are the first to be presented. This method was founded on a novel and non-conventional reasoning for demonstrative favorites called FO-MQCL (first order minimal qualitative choice logic). This tool is an improvement of the first order logic and a basic part of QCL. The QCL adds to classical preposition logic a new connective called ordered disjunction used to express preferences between alternatives. The approach entails in-modelling a security operator knowledge on systems or IDSs and his preferences so that alerts can be sorted and ones that fit operator security preferences are presented. This is a shortcoming because any knowledge and preferences beyond what is modelled in the systems and IDSs should not be reflected in the sense that some product of alert collection conditions can produce false evidence. The second shortcoming is the researchers overlooked the false positive alerts and false negatives alerts however the approach was able to reduce the redundant alerts.

Ghasemigol, *et al* (2015) concentrated on how to implement an intrusion alert correlation approach based on the evidence that occurred in the unprocessed alerts with no predefined expertise. The main idea was that the huge number of raw alerts contained some information that could be displayed by fewer hyper-alerts. The results were tested using the Darpa2000 dataset. The use of a trained dataset will not be able to capture all the alerts since attackers are using new methods and different sophisticated tools every time but is capable of eliminating the number of duplicated alerts created using several sensors although false alerts cannot be detected.

Elshoush (2014) proposed a state-of-the-art alert relationship structure aimed at restructuring the association modules to eliminate duplication, inappropriate and false alerts promptly to improve the productivity of the correlation procedure by reducing the correlation execution speed. According to (Elshoush, 2014) automation of the alert management and analysis is important because of the large number of false positive and irrelevant alerts and to enhance the investigation of these alarms therefore the alarms relationship are crucial for the purpose of investigating the alarms and generating the report of the status of network security. The researcher used the DARPA 2000 intrusion detection scenario specific datasets to evaluate the innovative alert correlation model. This is a trained dataset and will not be able to capture the new attacks.

Mohamed (2012) proposed a clustering algorithm referred to as hashing technique which eliminates performance issue. Comparable method can be implemented to provide a solution for the calculation and memory problem. The grouping system was validated based on DARPA dataset and a real time dataset from the attack monitoring unit. The proposed framework was not able to eliminate the dependency on human experts.

(GCT) to assess the correlation strength amongst alerts. Alert streams are modelled as time series. Granger Causality Index (GCI) is used to measure how much of the history of one-time series (the cause) is needed to correlate the evolution of the other one (the consequence or target). The frequency time series are built using a fixed size sliding-window. Compared to Maggi and Zanero (2007), they model the alert streams as random events rather than time series. They used a specific parameter to estimate the probability of the random alerts should be correlated Statistical models cannot reveal the dependencies among attributes and strongly depends on good choices of a parameter which proves to be both sensitive and difficult to estimated.

Joshi & Kakkar (2017) developed Honeypot based Intrusion Detection System that considerably enhanced discovery rate of IDS and significantly minimised false positives increasing the total productivity of the Intrusion Detection System. Honeypot based IDS significantly enhanced Average Throughput and Packet Delivery Ratio and also minimized the Energy Spent and Packet Drop Rate.

The investigator was capable of dealt with the limitations of IDS, both anomaly and signature detection by integrating them with the honeypot. The virtual Honeypot gathers evidence from IDS to promote monitoring if the traffic is suspicious or not. When the evidence is established to be susceptible then it is forwarded to the virtual honeypot else passed to its end application for execution.

The investigator delivered crossbred honeypot besides snooping agents to achieve optimum security contained by the wireless system. Honey pot are located next to the Firewall and intrusion machine intensely joined with snooping agents. Tracing is realized at packet level and pattern level of the traffic. However Jitter is not reduced which is undesired.

2.7 Machine Learning Techniques

Data mining is a helpful practice to uncover new insights, associations and hidden patterns within large data set of logs and messages (Parsaei *et al*., 2016 ; Kansra & Chadha, 2016). Knowledge Discovery in Database (KDD) practice is associated with extraction and discovery of useful information from large relational databases while data mining represents its core as decision support stage

The domain of Artificial Intelligence (AI) holds two methods to artificial learning (Dewa & Maglaras, 2016). The mental processes and artificial learning is inspired by the learning of algorithms alive in the human mind. The main aim (AI) is to learn how these algorithms can be interpreted into recognized vernaculars and computer courses. The practical computing standpoint is the second learning experience and has less grandiose aims. This encompasses building programs that acquire knowledge from past experience, and, hence, is a division of data processing. Machine learning deals with the model of learning and classification learning (Chakir *et al*., 2017).

Artificial Intelligence (AI) looks for methods and procedures to provide computers with human-like intelligence (Chalak, 2011). In the case of intrusion detection, because of the huge amount of data being processed in the cyberspace, it is required to use automatic tools that detect intrusions with minimum human intervention. Machine Learning (ML) is a branch of AI which provides such methods. ML algorithms automatically build detection engines from a set of events performing a training process(Gervais *et al.*, 2016). These models are then used to detect intrusions in real time. There are two classical approaches to train the system: supervised and unsupervised. In a supervised setting, the training dataset is labeled, and the learning algorithm knows to which class each trace belongs to. An unsupervised algorithm obtains a program that is able to separate traces from different classes without knowing which the exact class of each trace is. Clustering and Correlation-based algorithms are good examples of unsupervised ML. ML techniques offer the benefit that they can detect novel differences in traffic (which presumably represent attacks) by being trained on normal and attack traffic. The algorithms are discussed in greater lengths in subsequent subchapters.

1.8.5 2.7.1 Feature Selection

Feature selection techniques chooses accurate and significant features from intrusion dataset to have better results and less computational time. The prevailing alert correlation structures execute large volume of information that comprises of null values, less comprehensive information, and inappropriate features and hence manual analysis of the alerts is tedious, time-consuming and error-prone (Jama, Siraj, & Kadir, 2014; Siraj *et al.*, 2015). Feature set is selected from Correlation based Feature Selection (CFS) and those who were ranked high by the Information Gain (IG) measure based on a predetermined threshold. Correlation based feature selection selects the feature sets containing features highly correlated with class and uncorrelated with each other. Redundant features are removed because of high correlation with remaining feature set. Recent study indicates that machine learning algorithms can be adversely affected by extraneous and duplication of training and learning information (Noureldien & Yousif, 2016). Several algorithms like simple nearest neighbor algorithm is delicate to these extraneous characteristics, the optimum training features (its sample complexity) required to for a given correctness level increases exponentially as a result of the numbers of irrelevant attributes (Verma, 2016). The training features for decision tree algorithms can increase rapidly based on models including parity. The naive Bayes classifier, Decision tree algorithms such as C4.5 can be affected by redundant attributes. Bayesian classifiers are affected based on assumption that features are independent assigned the class (Juanchaiyaphum, Arch-int, & Arch-int, 2015) . (Madbouly, Gody, & Barakat, 2014; Manandhar, 2014) decision tree can over fit the training data, causing large trees and in most cases, pruning irrelevant and redundant information can result in C4.5 producing smaller trees.

The NIDS activates alert to the administrator to respond accordingly against the suspicious activities. Once an IDS finds a suspicious action, it immediately creates an alert which contains information about the source, target, and estimated type of the attack e.g., SQL injection, buffer overflow, or denial of service. As the intrusive actions caused by a single attack instance which is the occurrence of an attack of a particular type that has been launched by a specific attacker at a certain point in time are often spread over many network connections or log file entries, a single attack instance often results in hundreds or even thousands of alerts (Hofmann & Sick, 2011;Alhaj *et a*l., 2016). Most of the alerts generated are either false positive, like. benign traffic that has been classified as intrusions, or irrelevant, like. attacks that are not successful. The intrusion detection system should gather and analyze only the optimum features to accurately distinguish between normal and attack traffic with less computational time and resources usage. Feature selection should be incorporated into the existing intrusion detection to assist in selecting best relevance features subset which provides the best accuracy and removes distractions (Ramaki, Khosravi-Farmad, & Bafghi, 2015)s.

Feature selection also known as variable selection, feature reduction, attribute selection or variable subset selection, is a widely used dimensionality reduction technique, which has been the focus of much research in machine learning and data mining and has found applications in text classification, web mining etc. (Hasan *et al.*, 2016) . The feature selection techniques identify some of the important attributes that are appropriate in a data set to represent the attack steps by reducing the number of features, and removing irrelevant, redundant and noisy features. Eliminating unimportant features facilitates data visualization, improves modeling, prediction performance, and speeds up classification process. Dimensionality reduction, such as feature extraction and feature selection, has been successfully applied to machine learning and data mining to solve this problem. Feature extraction techniques attempt to transfer the input features into a new feature set, while Feature Selection (FS) algorithms search for the most informative features from the original input data enhancing the performance of alert correlation model. (Dewa & Maglaras, 2016).

For Feature Selection:

Feature selection procedures require four basic stages in a simple feature selection method (H. Singh & Kumar, 2015).

i. Generation procedure in order to generate the upcoming candidate subset
ii. Evaluation function so that it can evaluate the subset
iii. Stopping criterion to decide when to stop
iv. Validation procedure used for validates the subset

The existing feature selection techniques in machine learning can be broadly classified into two categories like wrappers and filters. Wrappers selection techniques calculate the value of features using the specific learning algorithm applied to the data while filters assess the value of features based on general characteristics of the data using heuristic algorithms. Feature selection algorithms can be further differentiated by the exact nature of their evaluation function, and by how the space of feature subsets is explored. Wrappers often give better results in terms of the final predictive accuracy of a learning algorithm than filters because feature selection is optimized for the particular learning algorithm used. However, since a learning algorithm is employed to evaluate each and every set of features considered, wrappers are prohibitively expensive to run, and can be intractable for large databases containing many features. Furthermore, since the feature selection process is tightly coupled with a learning algorithm, wrappers are less general than filters and must be re-run when switching from one learning algorithm to another.

The advantages of filter approaches in feature selection outweigh their disadvantages. Filters execute many times faster as compared to wrappers and therefore applicable in databases with a large number of features (Othman & Maklumat, 1999). They do not require re-execution for different learning algorithms and can provide an intelligent starting feature subset for a wrapper incase improved accuracy for a particular learning algorithm is required (Kumar, 2016). Filter algorithms also exhibited a number of drawbacks. Some algorithms do not handle noise in data, and others require that the level of noise be roughly specified by the user a-priori (Kumar, 2016; Song, 2016). In some instances, a representative of features is not chosen explicitly; however, features are ranked and the user subjectively chooses the final set. In other occasion, the user should stipulate the number of features required, or manually set the limits where the feature selection terminates. An algorithm may be designed such that it increases the initial number of features and result in a increase in the size of the search space (Song, 2016).

In contrast to the filter and wrapper models, the embedded model of feature selection does not separate the learning from the feature selection part (Song, 2016) . The embedded model integrates the selection of features in the model building. An example of such model is the decision tree induction algorithm Nevlud, Bures, Kapicak, & Zdralek, (2013); Shahadat, Hossain, Rohman, & Matin, (2017), in which at each branching node, a feature has to be selected. Another example of the embedded model are SVM-based feature selection methods Sivakumar & Srilatha, (2016), in which the task of feature selection can be understood as looking for the feature subsets that lead to the largest possible generalization or equivalently to minimal risk.

The major challenge is to obtain a feature set that is comprehensive enough to separate normal data from intrusion data and also keep the size of this set as small as possible. Typically, the more features in training and testing data set the more difficult is to solve problem. Increasing sub set of features, in machine learning will automatically increases the training time and consequently slow down run-time and memory requirements increase with more features. feature selection methods have been proposed by researchers within intrusion detection system to handle these kinds of problems.

In the research by Assi & Sadiq, (2017), five primary classification methods with three feature selection strategies have been implemented to classify the network attacks using NSL-KDD dataset. These methods are (J48 decision tree, Support Vector Machine (SVM), Decision Table (DT), Bayesian Network and Back Propagation Neural Network). The feature selection strategies are (Correlation base feature selection (CFS), Information Gain (IG) and Decision Table). Several experiments have been implemented to obtain good results using the training and testing NSL-KDD within general attack (Normal and Anomaly). These were carried out using four attack types: Denial of Service attack (DOS), User to Root attack (U2R), Remote to Local attack (R2L) and Probing attack. J48 classification method with information gain feature selection gives the best results (80.3%) using testing dataset and (93.9%) as an accuracy training dataset. Due to the rarity of U2R and R2L records and existing of the imbalanced dataset, detection of these classes by using conventional data mining approaches in intrusion detection became a challenging problem. The study by Parsaei *et al*., (2016) aims to improve the ability of intrusion detection systems in detecting

U2R and R2L attacks by exploiting SMOTE and creating a boundary margin for low frequency attack classes, coupled with the CANN technique, which is a combination of classification and clustering. In addition, the study utilized the NSL-KDD dataset. The forty-one (41) records are first reduced to a smaller dimensional set with 21 features using LOO method. Subsequently, in order to evaluate the proposed method, the dataset was sampled ten times by changing the seed of the random number generator. Furthermore, the number of U2R and R2L class instances were increased using SMOTE. The balanced dataset was then modeled by CANN and a single dimension dataset was extracted. At each execution of the algorithm, 10-fold cross validation was used for evaluations. Experimental results indicated that the proposed method outperforms the baseline approach regarding detection rate. However, it achieves lower accuracy and false alarm rate, which are not a significant difference. Results show that SMOTE coupled with CANN able to eliminate the limitation of the baseline research in detecting low-frequency attacks U2R and R2L and improves them by 94% and 50%, respectively.

2.7.1.1 Dimension Reduction Using Principal Component Analysis (PCA)

The number of features in a data set are is dimensionally condensed to reduce complexity and the resultant set is smaller compared to that of the initial dataset and the execution cost in anomaly based IDSs are expressively concentrated which makes them workable for real time deployment in high speed networks. The resultant data set can then be handled by the detection system which minimizes their complete computational cost without affecting their performances. The algorithms reduce massive data-set to a manageable size without significant loss of information represented by the original data (Syarif, Prugel-Bennett, & Wills, 2012). PCA is a mathematical technique that reduces and analysis data with several values and maps high dimensional data onto a lower dimensional subspace through removal of highly correlated and redundant features in the data without losing much of the information contained in the original dataset. The dimensionality reduced data components are statistically orthogonal to each other (I. Madbouly *et al.*, 2014). This enables speedup of training and robust convergence and hence can be applied in the intrusion alerts dataset to find the principal components of the alerts, like., the attributes vector that can describe the alerts exactly and sufficiently, but not redundantly.

Mathematically, the principal components of the distribution of the alerts, or the eigenvectors of the covariance matrix of the set of the alerts is established (Mallissery *et al*., 2014: Shahadat *et al*., 2017 : AliShah *et al*., 2015).

The computational complexity of the PCA is O ($p^2n + p^3$) p is the number of features and n is the number of data points. Covariance matrix computation is O(p^2n) and the corresponding eigenvalue decomposition is O(p^3). The dimensionality reduced dataset obtained after PCA are then analyzed by various classifiers namely, Naive Bayes, C4.5 decision tree, Support Vector Machine (SVM) and Multilayer Perceptron (MLP).

The feature selection techniques help to identify some of the important attributes in a data set, thus reducing the memory requirement, increase the speed of execution and improves the classification accuracy (Dewa & Maglaras, 2016). The following feature selection techniques have been used in data mining and machine learning techniques.

2.7.1.2 Correlation Feature Selection (CFS)

Correlation based feature selection (CFS) is considered as one of the simplest yet effective feature selection method which is based on the assumption that features are conditionally independent given the class, where feature subsets are evaluated based on an experimental evaluation function.(Wahba *et al*., 2015). A good feature subset is one that contains features highly correlated with the class, yet uncorrelated with each other. The major advantage of CFS, it is a filter algorithm which makes it much faster compared to a wrapper selection method since it does not need to invoke the learning algorithms. It is able to define the worthiness of an attributes by considering both redundancy amongst attributes and relevancy between features and class label (Barot, Singh Chauhan, & Patel, 2014; Shahbaz, Wang, Behnad, & Samarabandu, 2016). CFS has an added advantage over mutual information, mainly due to the computation of the PDF, for estimating the multivariate concentrations. Moreover, in case of any error mutual information, during the estimation reduces the efficiency of the feature selection technique.

in correlation significantly $$\operatorname{cov}(x,y) = \frac{\sum_{i=1}^{n}(x_i - \overline{X})(y_i - \overline{Y})}{n-1} \qquad \text{eqn 1}$$

where cov (X, Y) > 0 X and Y are positively correlated cov (X, Y) < 0 X and Y are inversely correlated

cov (X, Y) = 0 X and Y are independent

in which $\overline{X}$ are the x while $\overline{Y}$ mean values of y. The correlation coefficient p (x, y) varies from -1 to + 1 where

The closer to –1, the stronger the negative linear relationship

The closer to 1, the stronger the positive linear relationship

The closer to 0, the weaker any positive linear relationship

Pearson's correlation coefficient (2), where all variables have been standardized shows that the strength of the relationship between a compound and another variable is a function of the number of component variables in the composite and the magnitude of the inter-correlations among them, together with the magnitude of the correlations between the components and the outside variable. Subsequently, with an increase in the number of features, the correlation coefficient will increase due to their low correlation Shahbaz *et al*., (2016) and hence this technique might choice more number of features than is required to predict the class label. In addition, in practical communication set up, the dependence between network traffic data is not restricted to the linear correlations. Therefore, along with linear correlation, non-linear correlation should also be considered and another metric with the capability of analyzing the non-linearity correlation is required (Madbouly, 2016) . Thus, regardless of the type of the existing inter correlation among features and between features and class label, the system will be capable of finding characterizing features to the system behavior. Therefore, further processing is required to minimize the number of features in the candidate subset of features.

2.7.1.3 Information Gain

Information gain is used as a measure for evaluating the worth of an attribute based on the concept of entropy as shown in equation (1), +the higher the entropy the more the information content. Entropy can be viewed as a measure of uncertainty of the system. The largest mutual information between each feature and a class label within a certain group is then selected as shown in equation (2). The performance evaluation results show that better classification performance can be attained from such selected features (AliShah *et al.*, 2015; Barot et al., 2014).

Entropy is given by

$$-\sum_{i} P(c_i) \log_2 P(c_i).$$

Where p(ci) is fraction of examples in a given example eqn 2

The amount by which the entropy of the class decreases reflects the additional information about the class provided by the attribute and is called information gain (Nagle & Chaturvedi, 2013). Each attributes Ai itself and the class IGi is described in equation 3 below

$$IGi = H(C) – H(C|A_i) \qquad \text{eqn 3}$$

$$= H(A_i) – H(A_i|C)$$

$$= H(A_i) + H(C) – H(A_i|C)$$

Information Gain Algorithm

Input: A training dataset H = W (G, C), all the attributes to be selected

Output: Selected attributes z

```
1.  Set the factors: $P \leftarrow Pi$, i =1, 2, ...n, $D \leftarrow$'class labels', S =?
2.  for each attributes $Pi \in P$ do
        a.  compute the information gain IG(pi);
        b.  place pi into z in decreasing order with respect to IG(pi);
3.  Hold initial m attributes in z, then remove the rest;
4   Return Every attributes: z.
```

2.7.1.4 Chi-Square

Chi-square test is commonly used method, which evaluates features individually by measuring chi- square statistic with respect to the classes. It is a numerical test that calculates deviations from the expected distribution assuming the feature event is independent of the values in the class (Barot et al., 2014: Thaseen & Kumar, 2017).

$$\text{Chi square metric} = t\,(t_p\,(t_p + f_p)\,p_{pos}) + t\,(f_n\,(f_n + t_n)\,p_{pos}) + t\,(f_p\,(f_p + t_p)\,p_{neg}) + t\,(t_n\,(f_n + t_n)\,p_{neg}) \qquad \text{eqn 4}$$

The values are computed on metrics such as true positives (t_p), false positives (f_n), true negatives (t_n), false negatives (f_n), probability of the number of positives cases (p_{pos}) and probability of the number of negative cases (p_{neg}).

where t (count, expect) = (count – expect)^2/expect.

The technique follows the following procedure

i. Specify the hypothesis

ii. Device an analysis plan that determines how to accept or reject the hypothesis. the plan must specify

   a. Significance rank, the significance levels can be 0.001, 0.005 or 0.01 but it can be a value between 1 and 0

   b. The test method to test the independent level to identify whether there exists a relationship between two categorical attributes

iii. Examine sample data. The sample data should be analyzed to compute the degrees of freedom, predictable frequencies, test value and the P value associated with the test.

Degrees of freedom (DF) = $(r - 1) \times (c - 1)$

Where r is the number of levels of one categorical variable and c is the number of levels for other categorical variable.

Test statistics

$$\chi^2 = \sum_{i=1}^{k} \sum_{i=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}} \qquad \text{eqn 5}$$

Where,

k = No. of attributes,

n = No. of classes,

A_{ij} = number of instances with value i for attribute and j for the class,

E_{ij} = the expected No. of instances for Aij.

The larger value of the χ^2, indicates highly predictive to the class.

The important features are ranked based on performance according to a set of rules. The feature is removed at given time from the sample and the resulting sample is used for training and testing in the model. The procedure is as follows

i. Delete one input feature from the sample

ii. The resultant sample is applied as a training and testing sample in the model

iii. The results of the classifier are analysed based on the performance metrics

iv. The rules are used to rank the attributes by its importance level

v. Repeat the steps 1 to 4 for each attribute in the distribution sample

1.8.6 2.7.2 Unsupervised Machine Learning Techniques

In Unsupervised Machine Learning techniques, the alerts are grouped depending on the similarity of attributes. Similarity index or function is used to determine the degree of relationships. It can discover known group of alerts or attack steps, research by Alhaj *et al*., (2016); Siraj *et al*., (2009), claimed that the Structural-based alerts correlation model is unable to discover causal and statistical relationship, best suited for known attacks only. In contrast with supervised learning, in unsupervised learning there are no target output labels in the training and testing datasets *(Ambusaidi, He, & Nanda, 2015; Chand et al*., 2017). The algorithm receives inputs x_1, x_2,...x_n and the task is to learn and differentiate them. In unsupervised learning machine learn from the input dataset without knowledge about samples, such as normal and abnormal instances in network intrusion detection and learn the hidden structures inside the unlabeled data. The unsupervised learning algorithms can be categorized into: Dimensionality reduction and Clustering analysis.

Clustering is an unsupervised learning techniques which partitions unlabeled objects into meaningful subclasses such that members from the same cluster are quite similar and different to the members of different cluster (Kansra & Chadha, 2016;Kumar, Chauhan, & Panwar, 2013). A significant advantage of using clustering or unsupervised learning to detect network attacks is the ability to find new attacks or zero-day attacks. This indicates that attack types with unknown pattern signatures can be detected using this approach. Clustering results can also assist the network security administrator with labelling network traffic records as normal or intrusive. Examples of clustering techniques in machine learning includes K Means (Biswas et al., 2016; Nalavade, 2014; Solanki, 2014), Self-Organizing Maps (SOMS) (Smith, Japkowicz, & Dondo, 2008), Fuzzy C-means (FCM) (Amini, 2014; Mukosera *et al*., 2014), Expectation Maximization (EM) (Siraj, Maarof, & Hashim, 2009).

To increase the quality of alerts for investigation, some research in alert clustering for finding structural correlation have been done. The major problem in previous techniques is they relied heavily on Security Experts (SE) to develop and maintain the correlation system. The systems are based on already set rules or expert knowledge to maintain and analyze the intrusion alerts which, requires to updated regularly as the arrangements of these attacks changes frequently (Siraj, Maarof, Zaiton, & Hashim, 2011). Gorton, (2003) proposed an Aggregation and Correlation Component (ACC) groups of related alerts using a small number of relationships and the model is organized into four different layers: source, target, Probe layer and detailed target layer. The Correlation Component depend on a set of rules to group the alerts. Cuppens, (2002) proposed cooperative module for IDS while Ghorbani (2007) developed a Rule-Based Temporal ACS and both implemented a prerequisite and consequence knowledge-based data base to group and remove false positives alerts. These databases store establish logics employed as a support for discovering the relationship between incoming alerts and existing alerts. As a result, the two methods are time consuming and they need a huge amount of predefined and consequent rules in order to make necessary decision from the correlated alerts

Several attempts have been made to develop an intelligent supervised technique applicable in alert clustering. However, these approaches are time-consuming and they require regular setup and maintenance for their system. Cunningham (2002) introduced a hand-clustered algorithm which required a lot of alerts to be managed manually beforehand. Also, the system by Dacier (2002) required manual tuning periodically to convert network properties to support the clustering algorithm. These are some of the limitations that makes supervised learning-based correlation system less practical in the development of alert correlation frameworks. The work by Siraj *et al.*, (2009c) proposed a new hybrid clustering method called Improved Unit Range and Principal Component Analysis with Expectation Maximization (IPCA-EM), for alert aggregation in ACS. A major difference with this research is that the researcher employed Principal Component Analysis (PCA) to improve the system performance. Muchammad, (2015) proposed a new technique based on a recursive k-means clustering using Gini impurity index. The approach deals with two types of distance like: the distance whose value is the sum of data item to cluster centers and the distance whose value is sum of log distance from data to its cluster sub-centroids. The Localized k-nearest neighbor is used to minimize the number of training data in the classification phase. The experimental results obtained using KDD99 and Kyoto2006++ data set, indicated good performance in terms of accuracy and specificity, those are (99.57%, 99.75%) and (94.84%, 93.53%), respectively.

The unsupervised machine learning algorithms are applied when there is no class to be predicted but when the instances are to be subdivided into natural groups of instances determined by the features available to represent the items into clusters (Thaseen & Kumar, 2017). The algorithms can be trained on unlabeled data or can be applied to the test or evaluation data without training. The trained clustering algorithms build internal representation of unlabeled training data during training which apply to the test data set. The untrained clustering algorithms determines natural differences between subsets of data without prior insight into the data.

Most of the clustering techniques use some basic steps involved in identifying intrusion. These steps are as follows:

i. Find the largest cluster, like., the one with the most number of instances, and label it normal.

ii. Sort the remaining clusters in an ascending order of their distances to the largest cluster.

iii. Select the first K1 clusters so that the number of data instances in these clusters sum up to ¼ ´N, and label them as normal, where ´ is the percentage of normal instances.

iv. Label all the other clusters as attacks.

2.7.2.1 Self Organising Maps

The Self-Organizing Map Rathore & Jain, (2012) is a model that examine and provides a 3 dimensional data and it provides a very competitive learning model. It outlines a mapping from high dimensional input data space onto a regular two-dimensional array designed architecture as input vector with six input values and output is realized to two dimension spaces.

The SOM is a neural network trained with a competitive learning rule in an unsupervised manner. A competitive learning rule means that the neurons compete to respond to a stimulus, such as a connection vector (recall that a connection vector describes properties of a network connection, such as the destination port and number of packets sent). The neuron that is most excited by the stimulus, like. whose weight vector Parsaei *et al.*, (2016) is most similar to the connection vector, wins the competition. The winning neuron earns the right to respond to that stimulus in future, and the learning rule adjusts its weight vector so that its response to that stimulus in future will be enhanced, like. by moving the weight vector closer to the connection vector (Madbouly, 2016). This means that the next time that same connection vector is presented, the neuron that won the competition for that same vector last time will be more excited by it. During training, the SOM learns to project connection vectors that are close together in terms of Euclidean distance onto neurons that are close to each in the output grid. In this way, the SOM learns relationships between the connections a vector, expressing them as spatial relationships in the output grid. The training algorithm also ensures that the weight vectors of the neurons area good representation of the connection vectors in the training data. This is achieved by aiming for a low mean quantization error, where the quantization error is the distance between a connection vector and the winning neuron‘s weight vector. The mean quantization error is the average of this over all connection vectors in the training set(L. Li, Yu, Bai, Cheng, & Chen, 2018).

2.7.2.2 K - MEANS

The K-means algorithm, starts with k arbitrary cluster centers in space, partitions the set of the given objects into k subsets based on a distance metric. The centers of clusters are iteratively updated based on the optimization of an objective function. This method is one of the most popular clustering techniques, which are used widely, since it is easy to be implemented very efficiently with linear time complexity (Biswas *et al.,* 2016). The principle goal of employing the K Means clustering scheme is to separate the collection of normal and attack data that behave similarly into several partitions which is known as Kth cluster centroids. In other words, K-Means estimates a fixed number of K, the best cluster centroid representing data with similar behavior. The algorithm initially has empty set of clusters and updates it as proceeds. For each record it computes the Euclidean distance between it and each of the centroids of the clusters. The instance is placed in the cluster from which it has shortest distance. Assume we have fixed metric M, and constant cluster Width W. Let di (C, d) is the distance with metric M, cluster centroid C and instance d where centroid of cluster is the instance from feature vector (Biswas *et al.,* 2016).

K-MEANS ALGORITHM (Gambo & Yasin, 2017):

Input: The number of clusters K and a dataset for intrusion detection

Output: A set of K-clusters

Algorithm:

1. Initialize Set of clusters S. (randomly select k elements from the data)
2. While cluster structure changes, repeat from 2.
3. Determine the cluster to which source data belongs Use Euclidean distance formula. Select d from training set. If S is empty, then create a cluster with centroid as d.
4. else add d to cluster C with min (dist. (C, d)) or dist. (C, d) <=dist. (C1, d).
5. Calculate the means of the clusters. Change cluster centroids to means obtained using Step 3.

A distance function is required in order to compute the distance (like. similarity) between two items. The Euclidean distance is the most commonly used function and is defined as:

$$d\ (x, y) = \sqrt{}\sum (x_1 y_1)^2 \qquad \text{Eqn 6}$$

Where

x = ($x_1 \ldots x_m$) and y = ($y_1 \ldots y_m$) are two input vectors with m quantitative features.

In the Euclidean distance function, all features contribute equally to the function value. However, since different features are usually measured with different metrics or at different scales, they must be normalized before applying the distance function.

2.7.2.3 Fuzzy c-means (FCM)

Fuzzy c-means (FCM) is an improvement of K-means algorithm has become very important in field of intrusion detection system. The fuzzy C-means is a clustering technique that calculates the function relationship between individually test data occurrence and each cluster Amini, (2014). The experiment data occurrence is assigned to the group with higher membership. (Harish & Kumar, 2017). In fuzzy C-means, the individual data point can belong to several clusters at the same time. Nevertheless, the degree of membership is determined by membership grades which are assigned to each data point. For each x_i in dataset D the fuzzy C-means algorithm assigns membership grade u_{ij} which shows the degree of x_i membership in cluster j ($0 \leq u_{ij} \leq 1$). The membership grades are calculated for each example based on the minimization of an objective function which measures the distance between each data point and the cluster centers. let ***m*** be the size of the input dataset and ***K*** represents the clusters, this objective function is calculated as follows:

K membership value to each center. After that, it finds higher membership and assigns the instance to higher membership cluster. In other words, the instance in test dataset will divided into two clusters according to the degree of membership to C_1 and C_2 in this case. In the above equation q is the fuzziness exponent and can be any real value greater than 1 depending on the kind of problem. c_j is the center of j-th cluster and its dimensions are equal to that of input vector x_i. Creating the clusters is done through an iterative optimization process for objective function in which membership grades u_{ij} and cluster centers c_j are updated Once the Fuzzy C-means algorithm obtains the unlabeled dataset of magnitude m as input, it executes the above process and the output are two matrices: The Matrix U which consist of membership grades of each data example in each of the K clusters and matrix C which includes the cluster centers for K clusters (Sampat & Sonawani, 2015; Singh, 2013).

To create K disjoint subsets from the dataset based on matrix U, one subset for each individual example in the training dataset is determined based on its maximum membership grade like.

for each x_i: if $u_{iw} = \max \{u_{ij}\}$ then $x_i \in D_w$, where i = 1, 2, . . ., m; j = 1, 2, . . ., K.

After calculating the subset for all examples, the training dataset is divided to K disjoint subsets D1, D2, . . ., DK. These K subsets are used to train classification techniques like ANN, SVM etc.

2.7.2.4 Expectation and Maximization Algorithm (EM)

The EM algorithm Siraj *et al.*, (2009c) is a clustering technique in data mining and consists of two repeated steps, Expectation and Maximization. It is based on Gaussian finite mixtures model (GMM) for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables (N. Sharma, Bajpai, & Litoriya, 2012). The EM algorithms alternates between performing an expectation (E) step, which computes the expectation of the log- likelihood evaluated using the current estimate for the parameters, and maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step.

The model consists of a set of k probability distributions that represent the data of each cluster while the number of iteration and log likelihood difference between two iterations are parameters that defines each of the k distributions. Initially, the algorithm makes guesses for these parameters based on the input data, then determines the probability that a particular data instance belongs to a particular cluster for all data using these parameter guesses. The distribution parameters are revised again and this process is repeated until the resulting clusters have some level of overall cluster 'goodness' or until a maximum number of algorithm iterations are reached.

Mathematically, the algorithm attempts to find the parameters θ, that maximize the probability function, log P (x; θ) of the observed data. It reduces the difficult task of optimizing log P (x; θ) into a sequence of simpler optimization sub problems, whose objective functions have unique global maxima that can often be computed in closed form. These sub problems are chosen in a way that guarantees their corresponding solutions $\varphi^{(1)}$ $\varphi^{(2)}$ ... and will converge to a local optimum of log P (x; θ). The Expectation step (E-step) of the algorithm estimates the clusters of each data instance given the parameters of the finite mixture. During the E-step, the algorithm chooses a function $f(g_t)$, that lower bounds log P (x; θ) everywhere, and for which $f(\varphi^{(1)}) = \log P(x; \varphi^{(t)})$. The Maximization step (M-step) of the algorithm tries to maximize the likelihood of the distributions that make up the finite mixture, given the data. During the M-step, the algorithm moves to a new parameter set $\varphi^{(t+1)}$, that maximizes f (g_t). As the value of the lower-bound f (g_t) matches the objective function at $\varphi^{(t)}$, it follows (9), so the objective function monotonically increases during each of the iterations in EM.

$$\text{Log } P(x; \varphi^{(t)}) = g_t(\varphi^{(t)})\, g_t(\leq \varphi^{(t+1)}) = \log P(x; \varphi^{(t+1)}) \qquad \text{eqn 6}$$

1.8.7 2.7.3 Supervised Machine Learning Techniques

Classification is a supervised technique which takes each instance of a dataset and assigns it to a particular class (Urvashi & Jain, 2015 ; Assi & Sadiq, 2017). It extracts models defining important data classes. Such models are called classifiers (Gambo & Yasin, 2017). A classification-based IDS will classify all the network traffic into either normal or malicious. Data classification consists of two steps learning and classification. A classifier is formed in the learning step and that model is used to predict the class labels for a given data in the classification step. In analysis of Classification the end-user/analyst requires to know ahead of time how the classes are defined. Each record in the dataset already has assessment for the attribute used to define the classes. The main aim of a classifier Kaur & Sachdeva, (2016) is to explore the data to discover different classes and also to find how new records should be arranged into classes. Classification helps us to categorize the data records in a predetermined set and can be used as attribute to label each record and for distinguishing elements belonging to the normal or malicious class (Singh Arora, Kaur Bhatia, 2016). Different types of classification techniques are Random Forest, Bayesian networks, Support Vector Machine, Neural Networks.

The causal analysis finds the relationship between alert types in the alert stream to discover alert attributes that have the greatest impact on the relationship between intrusion alerts (Guha *et al*, 2016; Ramaki *et al*. 2015). The classification technique can discover unknown alerts but it is costly to develop a comprehensive attack database that can hold all the attack action with its preceding- and succeeding conditions (Siraj et al., 2015; Govindarajan, 2014;Song, 2016). Machine learning techniques learns through pattern and helps to reduce the sample set and it discovers novel types of attacks and extract attack steps automatically from labeled traffic data, thus overcoming the subjectiveness of human interpretation of intrusive behavior. Moreover, the intrusion detection can detect novel threats by learning through patterns.

AI-based Classification Thaseen & Kumar, (2016) is a machine learning technique where similar type of samples are grouped together in supervised manner and can classify the intrusion data as normal or attack. Several research employs and evaluated AI-based techniques adopting various classification techniques such as Support Vector Machines (SVM), Bayesian network Artificial Neural Network (ANN). The main challenge with the artificial intelligence techniques is that single-classification algorithm are not capable to discover all groups of attacks with satisfactory level of accuracy (Albayati & Issac, 2015; Panda *et al*., 2012; Tsai, Hsu, Lin, & Lin, 2009). Several current algorithms are local minima, for global minima, these techniques are expensive to compute; the prevailing techniques cannot correctly model postulate universe of problem; also the existing models are unstable in practical a case of neural networks that produces different outputs with various parameters due to the unpredictability intrinsic in the training varied systems trained from the similar data set might differ in their global performances and also may produce robust local differences process (Haddadi et al., 2010; Sunita et al., 2016; G. Wang, Hao, Ma, & Huang, 2010);. Every system may performs optimally within its own area in the feature space (Parsaei *et al*., 2016; Science, 2015). Similar estimation dataset challenges consist of insufficient volume of quality training data; the training data set contain imbalance features that affects the outcomes of classifiers to be inclined in the direction of majority class.

The concept of combining classifiers is proposed as a new direction for the improvement of the performance of individual machine learning algorithms (Li, et al 2018). AI-based ensembles learning models Khorshid *et al*.,(2015) , Chand *et al*., (2017) combines multiple and homogeneous, weak classifiers to solve advanced and complex problems and improve the classification accuracy of the final results. These models apply the same algorithm repeatedly through partitioning and weighting of a training data set and improves classification performance by the combined use of two effects like. reduction of errors due to bias and variance (Govindarajan, 2014). Adaptive hybrid systems have become essential in the field of soft computing and computational intelligence, the main reason being the high complementary of its components. The integration of the basic technologies into hybrid machine learning solutions facilitate more intelligent search and reasoning methods that match various domain knowledge with empirical data to solve advanced and complex problems. Application of collaborative and grouping of numerous estimates are mostly inspired by three characteristics which describe the intrusion detection field (Czarnowski & Piotr, 2016; G. Kumar & Kumar, 2012; Thaseen & Kumar, 2017): Appropriate evidence may be existing at several abstraction stages, the evidence may be gathered from manifold sources, and this evidence requires to be exemplified at the human level of understanding.

Artificial Intelligent (AI)-based techniques have been efficiently employed to improve the performance of algorithm in various areas such as business, bioinformatics, medicine, computer security, information science. Ensemble techniques helps solve several challenges in Intrusion detection system. The Ensemble combine multiple weak classifiers which supplement the limitations of each other as a result increase the complete performance. Ensembles applies the collective information to develop the assumption of the problem with various representative of dataset or attributes subspace therefore increase the performance even where insufficient volume of quality experimental data. Subsequently ensemble applies a combination of algorithms, hence it assists to discover the global solution which reduces the false alarms and increase the detection levels. The base algorithms help to produce the different set of base algorithms for effective ensemble. The Classifiers trained with similar dataset with different performance levels aid in maintaining diversity amongst the base classifiers(Kumar, 2016).

With the increased dependence on internet, Network Intrusion Detection system (NIDs) becomes an indispensable part of the information security system. NIDs aims at distinguishing the network traffic either normal or abnormal. Due to the variety of network behaviors and the rapid development of attack strategies, it is necessary to build an intelligent and effective intrusion detection system with high detection rates and low false-alarm rates. One of the major developments in machine learning in the past decade is the ensemble method that generates a set of accurate and diverse classifiers and combine their outputs such that the resultant classifier outperforms all the single classifiers. Several investigators implemented AI-based ensembles and hybrid approaches to improve performance of IDS. Their main aim is on the combination of classifying algorithms and associating the alerts to condense the false alarms and increase the detection accuracy for the security experts. combination of classifiers encompasses development of ensemble at initialization and selection learning stages, however ensemble integration stages encompasses grouping of different likelihoods of several classifiers (Panda et al., 2012; Sahu & Miri, 2017).

Chand *et al*, (2017) implemented a comparative study of the performance of SVM algorithm when it is weighted with additional classifiers like IBK, Bayes Net, Logistic, AdaBoost, J48, Random Forest, JRip, OneR and Simple Cart. The research concluded that Multi-Classifier algorithm performance is high as compared to an individual classification algorithm particularly when discovering attacks with low frequency such as guess password, rootkits, spyware etc. The introductory investigation conducted on NSL-KDD?99 dataset confirmed that stacking of SVM and Random Forest have the better performance with accuracy of approximate 97.50% and is relatively superior than SVM with 91.81%.

Assi *et al*, (2017) presented five primary classification methods with three feature selection strategies have been implemented to classify the network attacks using NSL-KDD dataset. These methods are (J48 decision tree, Support Vector Machine (SVM), Decision Table (DT), Bayesian Network and Back Propagation Neural Network). The feature selection strategies are (Correlation base feature selection (CFS), Information Gain (IG) and Decision Table). Several experiments have been implemented to obtain good results using the training and testing NSL-KDD within general attack (Normal and Anomaly). These were carried out using four attack types: Denial of Service attack (DOS), User to Root attack (U2R), Remote to Local attack (R2L) and Probing attack. J48 classification method with information gain feature selection gives the best results (80.3%) using testing dataset and (93.9%) as an accuracy training dataset.

Amini, (2014) proposed a novel ensemble method with neural networks for intrusion detection based on fuzzy clustering and stacking combination method. We use fuzzy clustering in order to divide the dataset into more homogeneous portions. The stacking combination method is used to aggregate the predictions of the base models and reduce their errors in order to enhance detection accuracy. The experimental results on NSL-KDD dataset demonstrate that the performance of our proposed ensemble method is higher compared to other well-known classification techniques, particularly when the classes of attacks are small.

Biswas *et al*, (2016)proposed that IDS with the amalgamation of best efficient features selected by Principal Component Analysis (PCA) can reduce the computational complexity of the system. It has been combined with the K-means clustering technique to cluster the specific groups of attacks and Artificial Neural Network to get a preeminent output by training the formulation of different base models. The model name has been defined by FP-ANK model. Investigational results have been reported on the NSL-KDD dataset where the accuracy rate associating with other models is distinct to validate the proposed system.

New hybrid classification methods was proposed by Govindarajan, (2014) for heterogeneous ensemble classifiers using arcing classifier and their performances are analyzed in terms of accuracy. Classification accuracy was evaluated using 10-fold cross validation. In the proposed approach, first the base classifiers RBF and SVM are constructed individually to obtain a very good generalization performance. Secondly, the ensemble of RBF and SVM is designed. In the ensemble approach, the final output is decided as follows: base classifier's output is given a weight (0–1 scale) depending on the generalization performance. The experimental results show that proposed hybrid RBF-SVM is superior to individual approaches for intrusion detection problem in terms of classification accuracy.

Different types of supervised standard, ensemble, and hybrid machine learning classification algorithms and models are introduced in research paper proposed by Khorshid et al., (2015) with the main focus on SVM classifier for prediction of the terrorist groups responsible of terrorist attacks in Middle East and North Africa from year 2004 up to 2008, by conducting different three experiments. The overall performance of the different types of classifiers used proved that hybrid machine learning classifiers perform accurately and in some situations, it could outperform the single classifiers with some enhancement, but ensemble methods are more accurate and outperformed the hybrid ensemble methods in their prediction of terrorist groups' attacks results.

2.7.3.1 Support Vector Machine (SVM)

The Support Vector Machine (SVM) is a supervised machine learning technique applicable in data mining for used for classification, outlier detection and regression. It is based on statistical learning theory and finds the optimal separating hyperplane between two classes , labeled pairs {(x, y)}, by determining a set of support vectors, which are members of the set of training inputs (Mukkamala, Janoski, & Sung, 1998). The Optimal hyperplane gives the maximum margin between training data set in the feature space. SVM converts the original data point into a dimensional space which are viewed as support vector in order to predict which class a new data point belongs to. There exist several hyperplanes in the feature space and the best hyperplane is the one with the largest margin separation between two classes while the distance from the nearest point on the each side is maximized (Akande, Owolabi, Twaha, & Olatunji, 2014) .

Support Vector Machine (SVM) provide a generic mechanism to fit the surface of the hyper plane to the data by the use of a kernel function (Pahwa, 2016). This gives an option for the user to provide a function e.g., linear, polynomial, or sigmoid during the training process, SVM selects support vectors along the surface of this function. The number of parameters used depends on the margin that separates the data points but not on the number of input features, hence SVMs do not require a reduction in the number of features in order to avoid over fitting (Ikram & Cherukuri, 2016). This is an advantage in the applications of SVMs in intrusion detection systems.

Support Vector Machine is a popular learning technique due to its high accuracy and performance in solving both regression and classification tasks compared to other techniques of classification (Shen et al., 2005). SVMs are moderately unresponsive to the number of data points hence the classification complexity does not depend on the dimensionality of the feature space, so they can potentially learn a larger set of patterns and thus be able to scale better than neural networks (Akande et al., 2014). when the data is organised into two classes, a suitable function can be applied to identify the most significant features, depending on the application.

SVM was originally designed for binary classification but practically, the problems can have multiple classes (Ikram & Cherukuri, 2016). The solution to this problem is to combine several binary classifiers although it increases the computational cost of SVM as the problem becomes complex. Thus the system could greatly reduce the training time and achieve better detection performance in the resultant SVM classifier. Shen et al., (2005) developed an SVM based intrusion detection system which combines genetic algorithm, hierarchical clustering algorithm and SVM technique. Sharma, (2016) developed intrusion detection system based SVM and ANN.

SVM Mathematically

i. Let training set $\{(\mathbf{x}_i, y_i)\}_{i=1..n}$, $\mathbf{x}_i \in \mathbf{R}^d$, $y_i \in \{-1, 1\}$be separated by a hyperplane withmargin ρ. Then for each training example $(\mathbf{x}_i, y_i)$:
   a. $\mathbf{w}^\mathbf{T}\mathbf{x}_i + b \leq -\rho/2$ eqn 7
   b. if $y_i = -1$ while $y_i(\mathbf{w}^\mathbf{T}\mathbf{x}_i + b) \geq \rho/2$
      i. $\mathbf{w}^\mathbf{T}\mathbf{x}_i + b \geq \rho/2$ if $y_i = 1$

ii. For every support vector $\mathbf{x}_s$ the above inequality is an equality. After rescaling $\mathbf{w}$ and b by $\rho/2$in the equality, we obtain that distance between each $\mathbf{x}_s$ and the hyperplane is

a. $$r = \frac{\mathrm{y}_s(\mathbf{w}^T\mathbf{x}_s + b)}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$$ eqn 8

iii. Then the margin can be expressed through (rescaled) **w** and b as:

i. $$\rho = 2r = \frac{2}{\|\mathbf{w}\|}$$ eqn 9

iv. Then we can formulate the *quadratic optimization problem:*

v. Find w and b such that is maximized and for all (xi, yi), i=1...n: yi (wTxi + b) ≥ 1

vi. Which can be reformulated as:

  a. Find **w** and *b* such that $\mathbf{\Phi}(\mathbf{w}) = \|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w}$ is minimized and for all ($\mathbf{x}_i$, y_i), i=1.n: y_i ($\mathbf{w}^T\mathbf{x}_i$+ b) ≥

The computational complexity of the support vector machine is O(nf$_{\text{eatures}}$· n$^2_{\text{samples}}$ () where n ^{2} is the number of data elements.

Compared with conventional machine learning methods SVMs have some

advantages (Sharma, Shrivastava, Lnct, & Bhopal, 2012; Thaseen & Kumar, 2017).

i. There are only two free parameters to be chosen, namely the upper bound and the kernel parameter.

ii. The solution of SVM is unique, optimal and global since the training of a SVM is done by solving a linearly constrained quadratic problem.

iii. Good generalization performance and Good robustness. Because of the above advantages, SVM has been recently used in many applications.

2.7.3.2 Random Forest

Random Forest is an ensemble learning technique for classification and predictive modeling. It is also an approach to data exploration and generates many trees by using recursive partitioning then aggregate the results (Pradhan, 2014). Each of the trees is constructed separately by using a bootstrap sample of the data and the bagging technique Pundir & Amrita, (2013) is applied to combine all results from each of the trees in the forest. The method used to combine the results can be as simple as predicting the class obtained from the highest number of trees.

Pseudo code

```
To generate c classifiers:
for i = 1 to c do
    Randomly sample the training data D with replacement to produce Di
     Create a root node, Ni containing Di
    Call Build Tree (Ni)
end for
Build Tree (N):

 if N contains instances of only one class then

return

else

    Randomly select x% of the possible splitting features in N

    Select the feature F with the highest information gain to split on

Create f child nodes of N, N1 ..., Nf, where F has f possible values (F1, …, Ff)

for i = 1 to f do

    Set the contents of Ni to Di, where Di is all instances in N that match Fi

    Call Build Tree (Ni) end for

end if
```

2.7.3.3 J48

J48 is an open source Java implementation of the C4.5 algorithm in the WEKA data mining tool (Pradhan, 2014). C4.5 is a program that creates a decision tree based on a set of labeled input data. The decision trees created from J48 is used for classification and hence C4.5 is a statistical classifier. J48 classifier algorithms Assi & Sadiq, (2017) are used to compare and built a decision tree using the information entropy process, from a set of training dataset. These algorithms adopt a top down technique and inductively built the decision tree for classification. It's extremely efficient when handling large datasets. (Gambo & Yasin, 2017). The extra features of J48 Dubb Shruti & Sood Yamini, (2013) includes accounting for missing values, decision trees pruning, continuous attribute value ranges and derivation of rules.

To make actual decisions regarding which path of the tree to replace is based on the error rates used. The reserved portion can be used as test data for the decision tree to overcome potential overfitting problem (reduced-error pruning).

Pseudo code

```
a. Check for base cases
b. For each attribute a
    a. Find the normalized information gain from splitting on a.
    b. Let a_best be the attribute with the highest normalized information gain.
    c. Create a decision node that splits on a_best.
    d. Recurse on the sub lists obtained by splitting on a_best, and add those nodes as
       children of node.
```

Now, among the possible values of this feature, if there is any value for which there is no ambiguity that is for which the data instances falling within its category have the same value for the target variable then terminate that branch and allocate to it the target value that have obtained.

2.7.3.4 Bayesian Network

Bayesian reasoning provides a probabilistic approach for inference and is based on the assumption that the quantities of interest are governed by probability distributions and that optimal decisions can be made by reasoning about these probabilities together with observed data (Ramaki *et al.*, 2015). A Bayesian network is a graphical model that encodes probabilistic relationships among variables of interest. Bayes' Theorem:

$$P(H \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid H) P(H)}{P(\mathbf{X})} = P(\mathbf{X} \mid H) \times P(H) / P(\mathbf{X}) \qquad \text{eqn 10}$$

When applied in combination with other statistical methods, Bayesian networks possess several benefits for data investigation (Kaur & Sachdeva, 2016;Assi & Sadiq, 2017). First, the Bayesian networks convert the relationship amongst variables and therefore they are employed circumstances where data are missing. Secondly, the Bayesian networks can develop a cause effect relationship between variables. Hence, they can to calculate the magnitudes of an action. Lastly, the Bayesian networks have both causal and probabilistic relationships; hence can be used to solve a problem by combining prior knowledge with data.

The Bayesian Classifier, or simple Bayesian classifier, works as follows (Noureldien & Yousif, 2016).

i. Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n- dimensional attribute vector, X = (x_1, x_2, …., x_n), depicting n measurements made on the tuple from n attributes, respectively A_1, A_2…, A_n.

ii. Suppose that there are m classes, C_1, C_2…, C_m. Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is, the naïve Bayesian classifier predicts that tuple X belongs to the class C_i if and only if. Thus, maximize P(C|X). The class Ci for which P(C|X) is maximized is called the maximum posteriori hypothesis.

$$P(C_i \backslash X) = \underline{P(X \backslash C_i)\ P(C_i)} \qquad \text{eqn 11}$$

iii. As P(X) is constant for all classes, only $P(X|C_i)$ $P(C_i)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, $P(C_1) = P(C_2) = \ldots\ldots = P(C_m)$, and would therefore maximize P(X|Ci). Otherwise, maximize $P(X|C_i)$ $P(C_i)$. Note that the class prior probabilities may be estimated by equally likely, that is, $P(C_1) = P(C_2) = \ldots. = P(C_m)$, and would therefore maximize $P(X_j|C_i)$. Otherwise, maximize $P(X|C_i)$ $P(C_i)$. Note that the class prior probabilities may be estimated by $P(C_i)=|C_i, D|/|D|$, where $|C_i, D|$ is the number of training tuples of class C_i in D.

iv. Given data sets with many attributes, it would be extremely computationally expensive to compute $P(X|C_i)$. In order to reduce computation in evaluating $P(X|C_i)$, the naïve assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the tuple (like, that there are no dependence relationships among the attributes).

The probabilities $P(x_1|C_i)$, $P(x_2|C_i)$ …., $P(x_n|C_i)$ from the training tuples. Recall that here X_k refers to the value of attribute A_k for tuple X.

The disadvantages of Bayesian networks includes (Kamesh & Sakthi Priya, 2014). First, the classification capability of naïve Bayesian networks is identical to a threshold-based system that computes the sum of the outputs obtained from the child nodes. Secondly, the child nodes do not interact between themselves and their output only influences the probability of the root node and hence incorporating additional information becomes difficult as the variables that contain the information cannot directly interact with the child nodes. Lastly, the accuracy of this method is dependent on certain assumptions that are typically based on the behavioral model of the target system and deviating from those assumptions will decrease its accuracy. Therefore, selecting an accurate model will lead to an inaccurate detection system as typical systems and/or networks are complex.

2.7.3.5 Artificial Neural Networks (ANN)

Artificial Neural Network (ANN) encompasses Multi-Layer Perceptron (MLPs) and Self Organising Maps (SOMs) which are the most application models in IDS. The Neural Networks (Kansra & Chadha, 2016; G. Kumar & Kumar, 2012), are supervised networks algorithms that transform input data into a desired response which are widely used for pattern classification. The ANN is composed of three components like. the neuron layers, the input layer, the output layer and the hidden layer. The input layer used in intrusion detection receives the input vector T from training dataset.

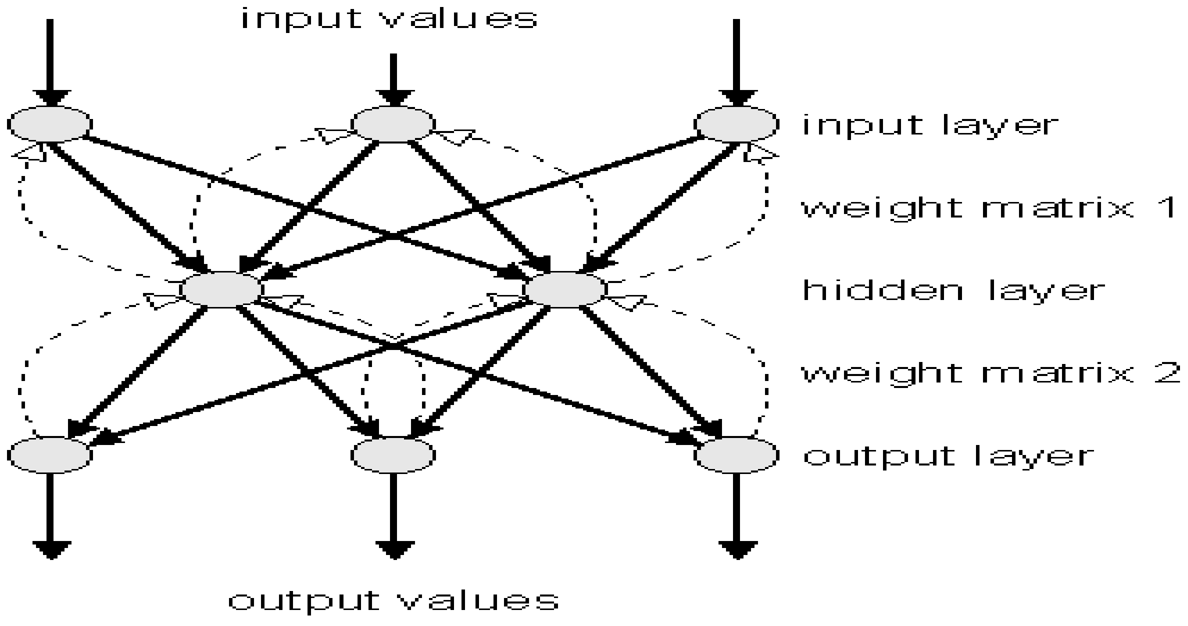


Figure 1: structure of artificial Neural networks

The input vector T has general format

$$T_i = \{t_1, t_2 ..., t_{ij} \quad \} \qquad \text{eqn 12}$$

Here, ij is the jth feature of ith instance of training/ test dataset. Total number of input neurons in input layer is equal to total features of training/test dataset for intrusion detection. The output layer contains the output neurons. The output neurons are equal to number of classes in dataset. A hidden layer is a middle layer. This layer adds a degree of flexibility to the performance of the ANN that enables it to deal efficiently with complex nonlinear problems. Each neuron in the single hidden layer receives the same input vector of N elements from the neurons of the

input layer, as defined by Equation (1), and produces the output. The input-output transformation in each hidden neuron is achieved by a mathematical non-linear transfer (or activation) function

$$Y_{ak} = f\{\sum_{i=1}^{n} wj\ k * t_{i\,j} + b_k\}..e \qquad \text{eqn 13}$$

Y_{ik} is the output of kth neuron in hidden layer for ith instance of dataset,

f () is activation function,

W_{jk} is the connection weight assigned to kth hidden neuron and jth neuron in input layer and k is the bias of kth hidden neuron. The activation functions

$$F(Y)= [[1+ \text{expr} [-\sum_{i=1}^{n} wj\ k * t_{i\,j} + b_k]]^{-1} \qquad \text{eqn 14}$$

The Neural Network architecture is based on the error-correction learning rule and approximate most problems with high accuracy and generalization ability (Kovac, 2012; Kidmose *et al.,* 2016). Error propagation is based on forward pass and backward pass. Feed forward networks allow signals between neurons to travel in one direction only; from input to output and mostly applicable in pattern recognition. Feedback networks allow for signals to travel in both directions, which in turns make them computational very powerful. However, it also tends to make them extremely complicated. Such a network functions dynamically, where the state of the network changes continuously until equilibrium is reached for the given input pattern.

Many researchers have successfully utilized ANN for IDS due to its advantages like (Kumar & Kumar, 2012; Amini, 2014; Aziz & Permana, 2015; Biswas, Shah, Tammi, & Chakraborty, 2016; Haddadi, Khanchi, Shetabi, & Derhami, 2010; Wang, Hao, Ma, & Huang, 2010).

i. High tolerance to noisy data;
ii. Ability to classify untrained patterns;
iii. Well-suited for continuous-valued inputs and outputs;
iv. Successful on a wide array of real- world data;
v. Algorithms are inherently parallel

However, there are several drawbacks of using ANNs:

i. in ANN large amounts of training data are necessary, and the performance of the network is directly dependent on this
ii. they are black boxes.
iii. determining the topology of the ANN is difficult and time consuming; mostly done ad hoc or optimized with an evolutionary algorithm.

2.7.3.6 IBK (K - Nearest Neighbor)

Instance based learners (IBL) are computationally simple and represent knowledge in the form of specific cases or experiences (Lakshmi, Prabakaran, & Ph, 2014) . IBL rely on efficient matching methods to retrieve stored cases so they can be applied in novel situations. Instance based learners are also called lazy learners because learning is delayed until classification time, as most of the power resides in the matching scheme. IB1 Chand et al., (2017) is an implementation of the k nearest neighbor based classifier where k is the number of neighbors. IB1 finds the stored instance closest according to a Euclidean distance metric to the instance to be classified and the new instance is assigned to the retrieved instance’s class.

$$D\ (x_j, y_j) = \surd \textstyle\sum_{j=1}^{n} f(xj, yj) \qquad \text{eqn} \quad 14$$

Equation 14 gives the distance between two instances x and y; x_j and y_j refer to the jth feature value of instance x and y, respectively. For numeric valued attributes f (x_j, yj) = $(x_j - y_j)^2$; for symbolic valued attributes f (x, y) = 0, if the feature values x_j and y_j are the same, and 1 if they differ.

```
i.   Input: KDD99 Testing Dataset U, KDD99 Training Dataset V
ii.  for each connection u from U
         a. for each connection v from V , calculate dist.=d (u, v)
iii. next
         a. sort v according to dist.
         b. select top k connections classify u as the majority of classes of the selected connections
iv.  next
```

The simple nearest neighbour algorithm is adversely affected by the presence of irrelevant features in its training data. While nearest neighbour can learn in the presence of irrelevant information, it requires more training data to do so and, in fact, the amount of training data needed (sample complexity) to reach or maintain a given accuracy level has been shown to grow exponentially with the number of irrelevant attributes. Therefore, it is possible to improve the predictive performance of nearest neighbor by removing irrelevant attributes. Furthermore, nearest neighbour is slow to execute due to the fact that each example to be classified must be compared to each of the stored training cases in turn. Feature selection can reduce the number of training cases because fewer features equates with fewer distinct instances especially when features are nominal. Reducing the number of training cases needed while maintaining an acceptable error rate can dramatically increase the speed of the algorithm(Syarif *et al*., 2012).

1.8.8 2.3.4 Ensemble Learning Techniques

The strategy in Ensemble classification systems is to create a set of accurate and diverse classifiers and combine their outputs such that the combination outperforms all the single classifiers. A classifier is accurate when its classification error is lower than that obtained when the classes are randomly assigned. Two classifiers are diverse if they make errors at different instances. classifiers ensembles are built in two phases: generation and combination. In the generation phase, the individual components of the ensemble, known as base classifiers, are generated. The techniques used to generate diverse classifiers are based on the idea that the hypothesis of a classifier depends on both the learning algorithm and the subset used to generate these classifiers. Three different approaches can be used to generate an ensemble of classifiers by varying the training set. Resampling the training examples is an approach applied by bagging and boosting for constructing classifier ensemble. Manipulating the input features achieve diversity between classifiers is by modifying the set of attributes used to describe the instances.

Manipulating the output target generates a pool of diverse classifiers with each classifier solving a different classification problem. The category that solve multiclass problems by converting them into several binary sub problems falls in this class. Methods that vary the learning algorithm can be subdivided in two groups like. approaches that use different versions of the same learning algorithm (homogeneous ensembles) and approaches where diversity is obtained using different learning algorithms (heterogeneous classifiers).

In the combination phase, the decisions made by the members of the ensemble are combined to obtain one decision. There are two main strategies for combining classifiers like. fusion and selection. Classifier selection presupposes that each classifier is an expert in some local region of the space. Therefore, when an instance is submitted for classification, the ensemble decision coincides with the decision given by the classifier responsible for the region of the space to which the instance belongs. In classifier fusion, the decisions from all members of the ensemble are combined in some manner to make the ensemble decision. Classifier fusion algorithms include combining rules, such as the average, majority vote, weighted majority vote, and the Board Count, and more complex integration models, such as meta-classifiers. A meta-classifier is a second-level classifier generated from the outputs given by the base learners. An ensemble classifier is a technique which combines multiple classifiers to improve robustness and also to improve classification performance from any of the constituent classifiers. Ensemble Learning Joshi & Srivastava, (2014) is a two-step decision making process, in which the first step is related to the decision of the individual classifier and the second step refers to the decision of the combined model. Ensemble methods create base classifiers and the outputs of each classifier obtained separately are combined, usually by voting, to get better classification accuracy.

Improved classification results can be achieved by using diverse classifiers such as Bagging, Boosting and stacking. Ensemble approaches Kulkarni, (2014); Pradhan, (2014) have several advantage over single model techniques like. the training and testing data source have no adequate data to select an individual and correct hypothesis, the learning practices for the weak classification algorithms can be defective, and the proposition space being investigated may not have the correct target function whereas the ensemble classifier can deliver a worthy estimation.

Ensemble Learning Joshi & Srivastava, (2014) is a two-step decision making process, in which the first step is related to the decision of the individual classifier and the second step refers to the decision of the combined model. Ensemble methods create base classifiers and the outputs of each classifier obtained separately are combined, usually by voting, to get better classification accuracy. Improved classification results can be achieved by using diverse classifiers such as Bagging, Boosting and Random Forest. The main goal of meta learning is to increase the understanding of how to improve performance of existing learning algorithms (Miškovic, 2014).

Recent research focuses more on how to improve the detection rates of machine learning classifiers. For example, Support vector machines Heba, Darwish, Hassanien, & Abraham, (2010), Bayesian belief networking Ramaki *et al.,* (2015), Artificial neural are investigated to model the IDS network (Amini, 2014; Aziz & Permana, 2015; Biswas *et al.*, 2016). They found that different classifiers performed better on different classes of intrusion. Also, by combining the best techniques for each class improved the overall performance of the detector. It seems that none was capable to discover all kind of malicious attempts professionally with high detection rate and low false alarm rate. Hence, the need to combine different classifiers as a fusion of machine learning method to enhance the detection accuracy of the model built in order to make efficient intelligent decisions in identifying the intrusions.

2.3.4.1 Bootstrap aggregating (bagging)

Bootstrap is a meta learning algorithm that improves classification and regression models in terms of stability and classification accuracy (Thomas & Balakrishnan, 2008; Science, 2015; Journal, Technological, Patel, & Tiwari, 2014). The algorithm takes bootstraps samples of objects and the classifiers are trained on each sample. The classifier votes are then combined by majority voting. A bootstrap sample is a statistical sample taken uniformly and with replacement, this means that the result sample set will contain duplicates (Kovac, 2012; Chaurasia & Jain, 2014;Sannady & Gupta, 2016). Given a training dataset of size N, Bagging creates M base models, each trained on a bootstrap sample of size N created by drawing random samples with replacement from the original training set.

```
BAGGING ALGORITHM:

INPUT: S: training set; T: no of iterations; n: bootstrap size

OUTPUT: BAGGED classifier:

H(x) = majority (h1(x)..., hT(x)) where ∈ [-1, 1] are the induced classifier

Given training data (x1, y1) ..., (in, yn)

For t=1, T:

For M bootstrap replicate dataset St by selecting n random examples from the training set with replacement

        let h(x) be the result of training base learning algorithm on St output combined
        classifier: H(x) = majority(h1(x)..., hot(x))
```

In the pseudocode for Batch Bagging in Figure 2., M is the number of base models to be learned, T is the original training set of N examples, Lb is the base model learning algorithm, the hi’s are the base learner model. A function h(x) is returned that classifies new examples by returning the class Y that gets the maximum number of votes from the base models h_1, h_2...... h_M

2.3.4.2 Boosting

Boosting is a machine learning meta-algorithm that built ensemble classifier by incrementally adding and iteratively learning weak classifiers with respect to a dataset to a final strong classifier (Amini, 2014). Bagging is better than boosting as boosting suffers from over fitting as it performs well only for the training data. Although equally they can increase detection accuracy significantly when compared with a single model, boosting have to a better detection accuracy. However, unlike bagging, boosting may also reduce the bias of the learning algorithm (Kumar, 2012).

Boosting algorithm (Chand *et al*., 2017)

Ad boost. A boosting algorithm which creates an ensemble of classifiers. Each one gives a weighted vote.

Input:

D, a set of d class-labeled training tuples;

k, the number of rounds (one classifier is generated per round);

a classification learning scheme.

Output:

A composite model.

Method:

```
a. Initialize the weight of each tuple in D to 1=d;
b. For i = 1 to k do // for each round:
    a. Sample D with replacement according to the tuple weights to obtain Di;
    b. Use training set Di to derive a model, Mi;
    c. Compute error (Mi), the error rate of Mi
c. If error (Mi) > 0:5 then
    a. Re initialize the weights to 1=d
    b. Go back to step 3 and try again;
d. End if
e. For each tuple in Di that was correctly classified do
    a. Multiply the weight of the tuple by error (Mi)=(1error (Mi)); // update weights
    b. normalize the weight of each tuple;
f. End for
g. To use the composite model to classify tuple, X:
    a. Initialize weight of each class to 0;
    b. For i = 1 to k do // for each classifier:
        i. $w_i$ = log 1error($M_i$)
        ii. error ($M_i$); // weight of the classifier's vote
```

```
            iii. $c = M_i(X)$; // get class prediction for X from $M_i$
            iv.  Add $w_i$ to weight for class c
      c. End for
      d. return the class with the largest weight;
```

2.3.4.3 Stacking

The Stacked Generalization is another method of merging numerous classifiers (Pradhan, 2014). Unlike bagging and boosting use a voting system, stacking combines several learning algorithms including decision tree, neural network, rule induction, naïve Bayes, logistic regression, etc. to generate the ensemble of classifiers. The Stacking produces an ensemble of classifiers in which, the base classifiers (level-0) are built using different training parameters. The results of every models are combined to form a novel dataset which is related to the real value that it is supposed to predict. Then, the stacking model learner as (level-1) use the output from base classifier to provide the final output.

Stacking tries to learn which classifiers are the reliable ones, using another learning algorithm the Meta learner to discover how best to combine the output of the base learners. The input to the Meta model also called the level-1 model is the predictions of the base models, or level-0 models. A level-1 instance has as many attributes as there are level-0 learners, and the attribute values give the predictions of these learners on the corresponding level-0 instance. When the stacked learner is used for classification, an instance is first fed into the level-0 models, and each one guesses a class value. These guesses are fed into the level-1 model, which combines them into the final prediction.

Stacking algorithm (Amini, 2014)

Input: Data set D = f $(x_1; y_1)$; $(x_2; y_2)$ …… $(x_m; y_m)$;

First-level learning algorithms L_1…… L_T;

Second-level learning algorithm L.

Process:

```
a. for t = 1…...T: $h_t = L_t(D)$    //Train a first-level individual learner ht by applying the first-level
b. end; // learning algorithm $L_t$ to the original data set D
```

```
c.  D'= ϕ; // Generate a new data set
d.  for i = 1…...m:
        a.  for t = 1…. T:
        b.  zit = ht(xi)          //Use ht to classify the training example xi
e.  end;
f.  D' = D' [ f (($zi_1$; $zi_2$; $zi_T$); $y_i$) g
g.  end; h' = L($D_0$). % Train the second-level learner $h_0$ by applying the second-level %
    learning algorithm L to the new data set $D_0$
h.  Output: H(x) = h' ($h_1$ (x) …… $h_T$ (x))
```

2.8 Data Fusion in Network Intrusion Detection

The concept of data Fusion was invented by the US Air Force project; the US Department of Defense initially anticipated a Joint Directors of Laboratories (JDL) model centered on national defense monitoring needs in 1987. Afterwards, JDL was progressively developed and applied in other sectors, such as automatic control, image recognition, target detection, and cyber security. The researchers have suggested the meaning of data fusion based on their fields of investigation. In order to evidently demonstration the role of data fusion expertise in network intrusion detection, an expression of data fusion is presented in this work. In general, DF can be applied into three layers according to where fusions are needed, namely, data layer, feature layer, and decision layer.

The lowest layer in the system is the data layer, it performs the function of processing and integrating raw network data; the feature layer is the middle layer that fuses and reduces features of the preprocessed data; the decision layer is the uppermost layer that merge and the interpretations of several processing components. In this area of network intrusion Detection systems, a lot of work concentrates on the feature layer and the decision layer. This is because the network data they need to fuse comes from public data sets that have already been fused at the data layer. The use of DF technology at the feature level can significantly decrease the magnitude of information processing, thus improving the productivity of NIDSs. Moreover, valuable and advanced information created by feature fusion can assist the decision-maker and additionally increase the strength and accurateness of the system. The use of data fusion expertise at the decision level, the decision fusion component combines the multiple decisions from local detectors to achieve new correct and dependable documentations of the network behaviors.

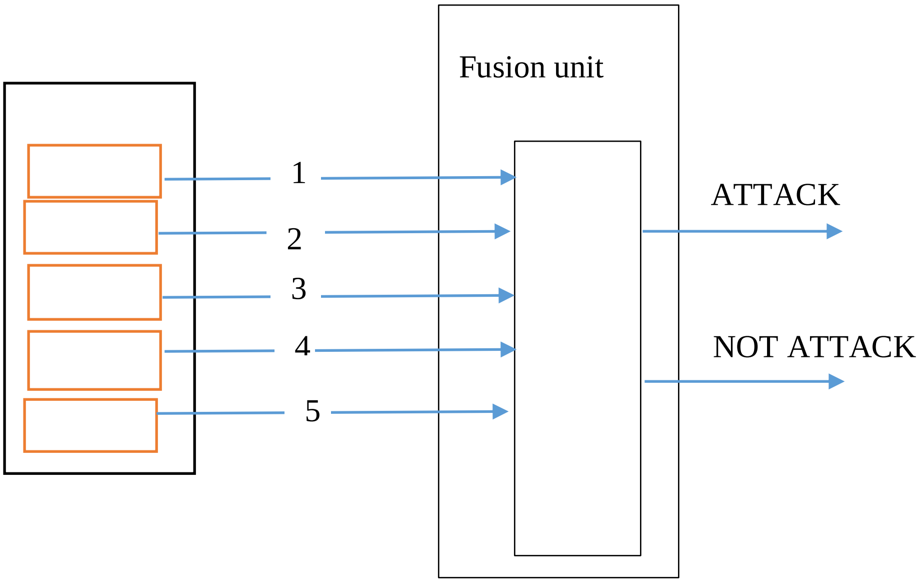


Figure 2: Fusion of Heterogeneous Classifiers for Network Intrusion Detection.

Multi sensor data fusion, or distributed sensing, is a comparatively novel engineering field applied to combine information from numerous and different devices and sources in order to create conclusions about events, actions, and circumstances. These arrangements are often associated with the human mental practices where the mind combines sensual information from the numerous sensual tissues, evaluates circumstances, creates conclusions, and directs action. This expertise has been in existence prominently in the military applications such as battleground, surveillance and strategic situation valuations. This technology has its application also in commercial applications such as robotics, industrial, health analytics, and remote sensing (G. Li & Yan, 2018).

The data fusion technology can be applied in technical fields that needs computational and experimental practices from areas such as statistics, Artificial Intelligence, operations research, digital signal processing, pattern recognition, cognitive psychology, information theory, and decision theory (Farid, Harbi, Ahmmed, Rahman, & Rahman, 2010). The application of multisensory data fusion in the field of IDS is based on mathematical theory

The application of data mining and data fusion techniques is in initial phases of technical development. Nevertheless, as computer networks is developing and as the cyberspace expands its of territories, the marketplace will drive ID systems toward next-generation capabilities. Integrated reasoning and decision-support tools are emerging requirements for robust and reliable intrusion detection in complex internetworks (R. Sharma *et al.*, 2012).

Data refinement is simplified when a common metalanguage for both intrusion detection and network work management exist. The temporal calibration of numerous streams of raw data from heterogeneous sources are also required. Internetworking protocols are evolving and may be used to synchronize objects and events in a distributed Internet environment. However, the security of TCP/IP information flows remain a critical issue. Correlation in physical space compares observations to a physical coordinate system (for example, the Euclidean distance between two measurements) to determine if there is a common source. Correlation in cyberspace requires the comparison of observations based on a different set of parameters such as source (IP address), network path, session flow, or behavior. The automated identification and tracking of dynamic intrusion subjects (suspected intrusion events) in cyberspace are also formidable technical challenges. For instance, intruders executing TCP-based attacks from numerous geographically dispersed net- works, or initiating attacks with one network connection and continues with another, sequentially changing IP addresses. Tracking and assessing the threat of these classifications of cyber attacks require new technical solutions (Zhang, Huang, Guo, & Zhu, 2011).

Classical Inference, Bayesian Inference, Dempster-Shafer Method, Generalized EPT, and Heuristic Methods are a few of the mathematical methods that are required in the decision-level identity fusion process. The adoption of these expertise in the field of intrusion detection and network monitoring is geared to recognize the situational awareness of cyberspace essential for progressive ID systems. Knowledge fusion which the uppermost level of interpretations is a very compound and inspiring area. For instance, imminent Intrusion Detection systems that detect and monitor several hostile evidences drifts for targets, attack rate, and severity in cyberspace. Defining the source of very sophisticated infiltrator in the cyberspace will continue to develop in complication as infiltrator become more cyberspace perceptive. The time allowable to the security administrator to monitor multiple attack sources can be expressed as the attack intensity and the probable situation assessment.

Thomas & Balakrishnan, (2008) have improved the detection accuracy of IDS by combining several IDS inputs. The weighting criteria in each IDS is combined to ensure accurate decision. DARPA 1999 data set which is obsolete is applied in evaluating the system. The dataset comprises of duplicated records, and hence it has an implication in classifier accuracy. In this research, binary values are applied to choose whether traffic is normal or abnormal. Giacinto & Roli, (2008) developed a pattern-recognition algorithm which combines several classification algorithms in network intrusion detection. It delivers an improved compromise amongst simplification capabilities and incorrect alarm generation.in contrast, the executions of fusion rules on unknown attacks indicated no enhancement above the outcomes of the single. The fusion rule delivers little enhancements on the accuracy of the neural network experimented on the complete feature set that achieves similar performance.

Vaughn & Bridges, (2004) developed the Decision Engine for an Intelligent Intrusion Detection System (IIDS). The systems combine information from unlike intrusion detection sensors based on an artificial intelligent method. Similar to neural networks algorithm it is unable to perform self-learning and self-training. It does not possess a technique for modifying the standard attack. Sharma, at el, (2012) proposed a novel method for intrusion awareness using data fusion and SVM classification.

Data fusion work on the bases of features gathering of event. Support vector machine is a super classifier of data for the detection of closed item of ruled based technique. The proposed method simulates on KDD1999 DARPA data set and get better empirical evaluation result in comparison of rule based technique and neural network model. Li & Yan, (2018) developed the evaluation standards for data fusion techniques for Network Intrusion Detection System. The performance of diverse data fusion methods is computed based on the proposed standards. The research concluded that, in the feature fusion technique, while adding to some outstanding fusion techniques, like SVM and MIFS, the enhanced categories of blending systems and hybrid fusion systems are usually effective and useable. In the decision fusion techniques, D- S Evidence Theory, NN, RF, and Ad boost are techniques that combines several decisions more accurately as compared to other methods from the research evaluated on KDD dataset series. moreover, the researchers establish several operative classification algorithms applied in NIDS, like, RF, C4.5, NN, and SVM, as well as their variants. However, the recent fusion techniques did not study the security and the scalability of data fusion.

To analyze the influence of security incidents on a networked system and accurately evaluate system security, Zhang *et al.*, (2011) proposes a novel cyber security situation assessment model, based on multi-heterogeneous sensors. By using D-S evidence theory, we fuse security data submitted from multi-sensors, according to the network topology and the importance of services and hosts. Moreover, they adopt the evaluation policy that from bottom to top and from local to global in this model.

The evaluation of a simulated network indicates that the proposed approach is suitable for network environment, and the evaluation results are precise and efficient. Zainal *et al*, (2009) demonstrated that ensemble of different learning paradigms can improve the detection accuracy. This was achieved by assigning proper weight to the individual classifiers in the ensemble model. Based on the experiment, LGP has performed well in all the classes except the U2R attacks. In contrary, RF shows a better true positive rate for U2R class. Thus, by including the RF in the assemble model, the overall performance particularly the result for U2R class has improved

There are three types of fusion techniques based on voting and includes the majority voting rule, the average rule and the belief function by considering how to combine decisions from basic classifiers (G. Li & Yan, 2018). The majority voting rule assigns a given input pattern to the majority class among the various outputs of the classifiers combined. They all contains computed values for individually classification algorithms and the concluding decision is based on the classification algorithm with the maximum computed value. where d = [$di_{,\ 1 \ldots}$, $d_{i,\ c}$]^T∈ {0,1}^{c}, i = 1⋯$L$is the outputs of the classifiers from the decision vector d, where L is the number of classifiers and $d_{i,\ j}$ =1is 1 or 0 depending on whether classifier i chooses j, or not, respectively. The last decision to combine several classifications algorithms is derived from the base classification output $d_{i,}$ (x) and corresponding weights b_i. This technique allocates a higher measure to the classification algorithm with the highest accurateness, nevertheless it disregards other erroneous base classification algorithms. The measures for the base classification algorithms are hard to attain and modify. hence, these algorithms are not capable to detect new network attacks. Naıve-Bayes, RF (Random Forest), Ada boost, and D-S evidence theories are classified as the type of winner-take-all. The average rule assigns a given input pattern to the class with the maximum average posterior probability, the average being computed among the K classifiers (this rule can be applied if classifiers provide estimates of posterior probabilities, like multi-layer perceptron neural networks). The third fusion rule is based on the computation of a "belief" value for each data class given the set of outputs of the K classifiers. Belief values are based on estimates of the probabilities that a pattern assigned to a given data class actually belongs to that class or to other classes. These probabilities can be easily computed from the confusion matrix on the training set. The classification is then performed by assigning the input pattern to the data class with the maximum belief value.

Several researchers combine the feature selection and classification algorithms to improve the detection accuracy and make intelligent decisions in determining intrusions. Siraj, Maarof, & Hashim, (2009a) developed a novel, computerized and knowledgeable model by combining several clustering algorithms clustering. The model called Improved Unit Range and Principal Component Analysis with Expectation Maximization (IPCA-EM) is capable of clustering related alerts and to filter the low quality alerts. Panda *et al.*, (2012) developed a hybrid intelligent method based on combination of classifiers in order to make the decision intelligently, so that the overall performance of the resultant model is enhanced. These two models use hybrid classifiers to make intelligent decisions and the filtering process is applied after adding supervised or unsupervised learning techniques to obtain the final decision. Shah, Qian, & Mahdi, (2017) developed a hybrid technique for anomaly intrusion detection system using a blending of both entropy of essential network features and support vector machine.

Madbouly *et al*. (2014), proposed a relevant feature selection model that selects a set of relevant features to be used in designing a lightweight, efficient, and reliable intrusion detection system. Although, the model achieved good overall detection result; detection results for PROBE, U2R, R2L attack types were low.

Lin *et al*. (2015) studied the importance of feature representation method on classification process. They proposed cluster centre and nearest neighbour (CANN) approach as a novel feature representation approach. In their approach, they measured and summed two distances. The first distance measured the distance between each data sample and its cluster center. The second distance measured the distance between the data and its nearest neighbour in the same cluster. They used this new one-dimensional distance to represent each data sample for intrusion detection by a k-nearest neighbour (k-NN) classifier. The proposed approach provided high performance in terms of classification accuracy, detection rates, and false alarms. In addition, it provided high computational efficiency for the time of classifier training and testing. Zhao et al. (2015) proposed a new model based on immune algorithm (IA) and BPNN. The new developed method is used to improve the detection rate of new intruders in coal mine disaster warning internet of things. IA was used to preprocess network data, extract key features and reduce dimensions of network data by feature analysis. BPNN is adopted to classify the processed data to detect intruders. Experiments' results showed the feasibility and effectiveness of the proposed algorithm with a detection rate above 97%.

2.9 Discussion and Analysis of Alert Correlation Techniques

1.8.9 2.9.1 Alert Correlation Techniques Comparison

All the discussed techniques have their advantages and disadvantageous as summarized in

Table 1 Alert Correlation Techniques Comparison

Technique	Advantages	Disadvantages
Similarities of Alert Attributes technique	Can reduce large number of redundant alert generated by multiple sensors. (Siraj et al. 2015; Yusof, at el 2008)	Suitable for known alerts. Not able to discover causality of alerts and statistical relationships. Limited to discover complicated attacks (Alhaj *et al.*, 2016).
Predefined Attack Scenario	Is able to accurately detect well-documented attacks. Can reduce large number of redundant alert generated by multiple sensors. (Chahira, Chuka, & Kemei, 2016; Chakraborty, 2013).	Could generate large number of false pose amative alarm. It requires that users specify attack scenarios manually. It is limited to detection of known attacks or misuse detection and not anomaly detection. Multi-step attack alert is disregarded intrusion, (Chahira *et al.*, 2016; Shameli Sendi 2013; Siraj *et al*,. 2015).
Prerequisites and Consequences of	Multi-step attack can be detected to provide a high-	The approach may not be practical in production networks due to the

Attacks	level view of the attack associated with a security compromise. Can generate useful graph to determine the attacker's objective.	complexity of the design and user behavior(Siraj *et al.*, 2015). It is expensive to build a complete attack database which consists of every attack action with its pre- and post-conditions (Kamesh *et al*.2014; Siraj *et al*. 2015; Wang, 2010).
data mining and machine learning	Does not need pre-defined knowledge about attack scenarios. Using anomaly detection technique. New attack scenarios can be identified. Can be used as pre-process alerts or meta-alert signatures.	Avery huge computational overload especially in large scale networks. This approach requires a lengthy initial period of training (Dewa,*at el*, 2016; Madbouly *et al*. 2014; Juanchaiyaphum, *et al.* 2015; Shahadat *et al*. 2017).
Hybrid technique	Performs multiple types of correlations (structure, cause & statistical). No predefined rules. Recognize known and unseen alerts. No manual parameters settings(Fanfara, *et al.*, 2013; Rasmi *et al.* 2015; Science 2015).	May lead to complex architecture (Chahar *et al.* 2017; Juanchaiyaphum, *et al* 2015).

1.8.10 2.9.2 Problems in existing Alert Correlation Systems

Based on the survey and analysis conducted on various alert correlation systems, there is currently no silver-bullet solution to the AC problem. The problems in alert correlation can be grouped into three constructing attack scenarios, improving qualities of alerts, and dataset and validation. In each group, the investigation problems are recognized and emphasized in the following sub-sections.

i. Constructing Attack Scenarios

Most of the current Intrusion Detection Systems (IDSs) adopts a single classifier algorithm to classify the network traffic as normal behavior or anomalous data (Scholar, 2017 ; Miškovic, 2014 ; Sunita, Chandrakanta, & Chinmayee, 2016). However, these single classifier systems fail to identify various category of attacks competently in relations to high detection rate and low false alarm rate. In combination with that, the other challenge is in what the alerts should relate with one another so that they can create association to discover the step-by-step attack setups (Guha, 2016; Roschke, Cheng, & Meinel, 2010) . Consequently, system analyst will recognize the previous step by step attacks which been launched in order to confirm if the subsequent attack is successful.

ii. Improving qualities of alerts

The existing NIDS handles large volumes of information that comprises of worthless values, inadequate information, and irrelevant features causing the analysis of the alerts to be tedious, time-consuming and error-prone

In several situations an intrusion varies from normal events only slightly, occasionally even only the setting in which the action happens determines if it is an intrusive. Due to harsh real-time conditions, IDSs doesn’t evaluate the context of entirely events to the level necessary (Shen *et al*., 2005; M. M. Siraj & Hashim, 2008). Scripting signatures for operating systems is a very difficult task (M. M. Siraj & Hashim, 2008). In other situation, the correct equilibrium between an excessively exact signature (which is cannot capture all attacks or their variations) and an overly general one (which recognizes legitimate actions as intrusions) can be difficult to determine. Activities that are regarded normal in some settings might be malicious in others (Siqueira, Ruiz, & Loureiro, 2014). For instance, carrying out a system scan is malicious except if the system performing it has been approved to do so. The lOSs deployed through the normal out-of-the box conformation will most likely identify several normal actions as malicious.

iii. Datasets and Validation

Currently, there exist no standard performance assessment approach exists for ACS (Nadiammai & Hemalatha, 2014 ; Assi & Sadiq, 2017; Govindarajan, 2014). Also, no dataset clearly developed for evaluating alert correlation algorithms is publicly obtainable. The extent such a dataset can be created and authenticated is an open investigation problem as well. The NSL-KDD dataset Jain & Rana, (2016); Parsaei *et al.,* (2016); Shahadat *et al*., (2017) which is derived from original KDD-99 and has eliminated some of its drawbacks is analyzed. The simulated attacks in the NSL-KDD dataset fall in one of the following four categories Denial of service attack (Dos), Probe attacks, Remote-to-Local (R2L) attacks, and User-to-Root (U2R) attacks.

Many standard data mining process such as data cleaning and pre-processing, clustering, classification,regression, visualization and feature selection are already implemented in WEKA (Verma, 2016). The automated data mining tool WEKA was used to perform all these experiments on the 20% NSL KDD dataset. The performance metrics used to evaluate the proposed network intrusion detection system are: the speed of the model, the classification accuracy and the false alarm. The classification accuracy of an intrusion detection system is measured by the precision, recall and F −measure; which are calculated based on the confusion matrix. In all experiments in this thesis, 10-fold cross validation training and testing mode was used because it reduces the variance of estimate. In 10-fold cross validation training and testing mode, the data is randomly divided into 10 parts in which the class is represented in approximately the same proportions as in the full dataset (Dewa & Maglaras, 2016; K. Kumar, 2016).

Each part is held out in turn and the learning scheme trained on the remaining nine parts; then its error rate is calculated on the holdout part. Thus, the learning procedure is executed a total of 10 times on different training sets (each set has a lot in common with the others). Finally, the average of 10 error estimates is calculated to obtain an overall error estimate. Extensive tests on numerous different datasets, with different learning techniques, have shown that 10 is about the right number of folds to get the best estimation of error, and every subsample is used for both the training and testing. All experiments are performed using Windows platform with the following configuration Intel Core-i5 processor, 2.5GHz speed, and 8GB RAM. As already explained in the approach there are some certain things that has to be done before the data set is ready to be used by the algorithms. Network intrusion detection systems deal with a huge amount of data that contains null values, incomplete information, and irrelevant features. The analysis of the large quantities of data can be tedious, time-consuming and error-prone. Data preprocessing Aims at Cleaning the data to remove noise and duplicate information and then deal with any incomplete or missing data. There are multiple conversions and modifications that needs to be done, in order for the algorithms to run properly and without error due to the data set. The data input format for WEKA is an "arff" file, with "arff" being the extension name of your input data file. WEKA can also read from CSV files and databases. The following two preprocessing stages has been done on NSL-KDD dataset: Mapping symbolic features to numeric value based on discretization. Implementing scaling since the data have significantly varying resolution and ranges based on normalisation. The attribute data are scaled to fall within the range [-1, 1]. Attack names were mapped to one of the five classes, 0 for Normal, 1 for DoS (Denial of Service), 2 for U2R (user-to-root: unauthorized access to root privileges), 3 for R2L (remote-to-local: unauthorized access to local from a remote machine), and 4 for Probe (probing: information gathering attacks)

2.10 Alert Correlation Design Considerations

An AC framework may consist of several tasks: normalization, reduction, severity/prioritization, attack detection and prediction to provide a high-level view of network security situations. These goals require a framework that effectively, efficiently and accurately deals with the massive alerts. Predicting the next actions of the attacker is very important and difficult task. Prediction helps intrusion prevention systems reacts properly before the network is compromised and gives the opportunities to overcome the advantages of attacker. However, existing works on overcoming the limitation of NIDSs (in term of producing high volume of low-quality alerts) neither deals with Alert Correlation nor Attack Prediction as a proactive approach. Several intrusion alert correlation techniques have been reviewed and analyzed to identify the design consideration and the proposed solution in order to improve the IDS problem as discussed below.

Formatting the alerts can be considered as an important initial task in the preprocessing task of AC framework. Nowadays, the majority of organizations implement different types of NIDSs (heterogeneous NIDSs), accordingly they produce alerts in different data format. Alert normalization Thaseen & Kumar, (2016) is a process to convert different alert data formats from multiple intrusion sensors into a standard format to be appropriate and acceptable by the other correlation components. Information Detection Message Exchange Format (IDMEF) Alhaj *et al.*, (2016); Siraj *et al.*, (2015); Wang, (2010) data model is a standard representation of alerts into a class with the following set of attributes: {Alert ID, Sensor ID, Timestamp, Source IP Address, Source Port, Destination IP Address, Destination Port, Service Protocol, Alert Type}. The benefits of IDMEF can be expressed in terms of adaptability and transparency. IDMEF is adaptive since its operating tractability on any NIDS design so long as the alarms generated holds valuable evidence to the Network Administrator. IDMEF as well is transparences in advancement procedure. The configuration in IDMEF can be improved and influenced easily based on Network Administrator requirements and conditions as it is constructed on extended markup language (XML). Using the IDMEF-based alerts, the features can be visualized and mined easily but, IDMEF cannot address the problem of alert redundancy where NIDSs produced multiple repeated (similar) alerts in a short duration of time. Such redundant alerts can overload and may contribute to false correlation. Therefore, they must be identified and reduced.

IDS are susceptible to alert saturating for example they provide a huge amount of alerts to the security administrator, who in turn has the complications handling the load (Smith *et al.*, 2008; Zhou, Leckie, & Karunasekera, 2010). To reduce number of alerts generated from IDS and improve the alert correlation performance in regard to the execution time and quality of alerts by adopting alert filtration and alert aggregation. Aggregation of alerts that are generated by same NIDS or different NIDSs usually belong to the same attack and they are identified by the same source and target IP addresses and blended with repeated/redundant alerts. Such case increased the number of alerts and produce high-dimensionality of alerts (Hajamydeen *et al.*, 2016; Rasmi & Al Qerem, 2015). In practice, the redundant alerts are usually false positivism and aggregation is used to group repeated or redundant alerts and are represented as one meta-alert or hyper alert. Although the filtration of false positives and aggregation of repeated alerts can improve the alerts quality, the hidden useful meaning contains in the alerts is still unrevealed (Das, Pathak, Sharma, & Srikanth, 2010; Siraj *et al.*, 2015). Thus, extraction of meaningful information from the alerts can be achieved by recognizing the attack scenario.

Constructing attack scenario is important and crucial in AC research to study the behavior of the attacker (Siraj *et al.*, 2010; Siraj *et al.*, 2015). It is challenging because alerts contain low level information. In practical, attack scenario should consist of a number of attack stages, and an attack stage should contain a list of attack steps. Therefore, in order to recognize attack scenario, two problems need to be addressed: 1) Identifying Attack Steps, and 2) Recognizing Attack Stages. Common pattern of alerts should bring useful information. Finding the commonalities among group of alerts is the problem of identifying the common attack steps. This problem can be solved by clustering/grouping common alerts based on the similarities of certain or all attributes. They are two issues need to be considered: i) How to define and determine the level of similarity, and ii) How to group unknown/new alerts. In determining level of similarity, some researchers for example used a predefined similarity probabilistic-based function to measure similarity between two alerts. Grouping the unknown or new alerts can be achieved by using unsupervised machine learning algorithms *(*Alhaj *et al*., 2016; Urvashi & Jain, 2015). The work by has shown that grouping similar attributes not only reveals the attack steps, but it can reduce a large a number of alerts as well. Even though clustering can effectively correlate some alerts, it cannot discover the causal relationships between alerts and hence recognizing attack stages is essential to discover the causal relationships (Czarnowski *et al.*, 2016 ; Madbouly, 2016; Song, 2016).

Recognizing attack stages are closely related to a classification problem because it attempts to classify the alerts into the corresponding cause/class and based on the cause-effect paradigm, derived rules and knowledge on the known attack stages to construct the cause and effect of an attack stage (Assi & Sadiq, 2017; Dhanabal & Shantharajah, 2015; Kaur & Sachdeva, 2016; S. Kumar & Naveen, 2016; Yang, 2011). As the patterns of intrusion changes, the classification should flexible enough to permit the introduction of new alerts where their properties may belong to neither class nor several classes. Training-testing paradigm using supervised machine learning algorithms is more practical although if using unlabeled dataset, the labeling of target attribute for data training needs to be done beforehand

Not all generated alerts are equally important in terms of their severity and critically of the target being attacked, so there is need to separate few important alerts from the rest and give them priority (Valeur, 2006). Work by Shameli-Sendi & Dagenais, (2014) categorized severity of alerts into three types: low, medium and high. High severity alerts are always referred to high risk alerts that can cause huge damage to network assets. They used information in the NIDS signature files to identify the type of severity. Normally, alerts that are low severity will be ignored by security analyst for future correlation process. Shameli-Sendi & Dagenais, (2015) proposed a fuzzy-logic based technique for scoring and prioritizing alerts. Their method evaluates alerts based on a number of criteria and used Fuzzy logic inference mechanism in order to score/prioritize alerts.

NIDS technologies play a vital role in protecting communication networks against cyber-crime. However, these technologies are not very effective in predicting the future attacks as they generates alerts when attack activities/intrusions have taken place (Chatur, 2014; Djemaa & Okba, 2012). A proactive approach Iafarov, Gad, & Kappes, (2015); Siraj *et al.*, (2015) is to anticipate and conduct possible attacks to prevent damage. Accordingly, the next step of an attack can be predicted after detection of few steps of attack in progress, so predicting the next actions of the attackers are an important and difficult task. Attack Prediction Rasmi & Al Qerem,(2015); A. Singh, (2009) can help intrusion prevention systems reacting properly before the network is compromised having the opportunities to overcome the advantages

Alert visualization provides a way of alert analysis and presentation to the network administrator can either be text based or graph based (Roschke, Cheng, & Meinel, 2010; Yusof *et al.,* 2011). Graph based provides a visual way of presenting and analysis which is easier and less error prone. The efficiency of a module depends greatly on the nature of the data-set analyzed to evaluate the system. Different data-sets Kang, (2015); Khorshid et al., (2015); Nagle & Chaturvedi, (2013) will yield dissimilar worth of outputs in diverse correlation components. Most common data set include DARPA 1999 and DARPA 2000 latest attack scenario data-sets to include IPv6 attack to confirm its effectiveness and success in generating a better and quality output.

2.11 Conceptual Framework

Based on comparison of alert correlation techniques, alert correlation design considerations and analysis done in the previous section, the proposed framework consist of four main stages, feature selection structural correlation, causal analysis, performance evaluation.

The framework receives regular and anomalous traffic pattern as input and performs feature selection to select the appropriate features for alert correlation. Several feature selection approaches (correlation based and information gain) will be evaluated to discard the redundant and irrelevant features from the training and testing dataset. The selected features are preprocessing using normalization and discretization techniques. Dimensional reduction technique based on Principal Component Analysis (PCA) identifies a suitable low-dimensional representation of original data. Reducing the dimensionality improves the computational efficiency and accuracy of the data analysis. The output is sent to a clustering technique to group alerts together arising from the same event and merge the cluster as hyper alerts. A causal analysis module based on integration of several classifications (Bayes, Functions, Rules and Trees Classifiers) techniques are used to classify the normal and anomalous instances of our dataset. The output from each classifier proceed into the decision unit, and the overall decision is derived based on the majority voting rule. If majority outputs from classification unit recommend Attack, hence the decision unit concludes that the input traffic is of ATTACK type; else it is NOT ATTACK.

The performance evaluation is done based on classification performance, time consumption and resource consumption.

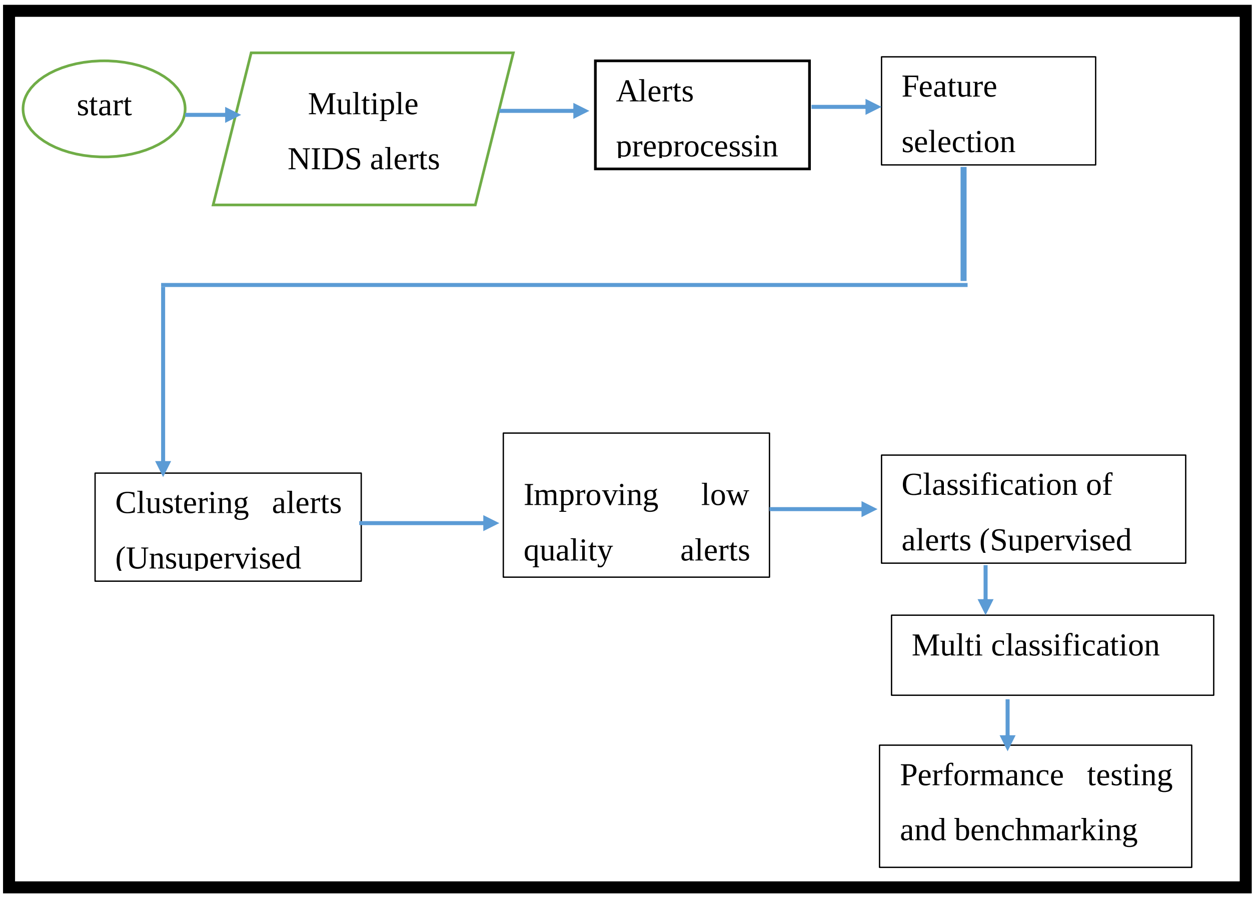

start
Multiple NIDS alerts
Alerts preprocessin
Feature selection
Clustering alerts (Unsupervised
Improving low quality alerts
Classification of alerts (Supervised
Multi classification
Performance testing and benchmarking

Figure 3: Conceptual framework

2.12 Conclusion

This chapter found that the main problems in AC to be addressed are improving the alerts quality and recognizing attack strategy. These problems have been addressed separately due to restriction of the applied AC method that can only offer single type of correlation. As a result, the correlation offered by the existing single based AC models is not optimal and incomplete. Most of the related works are incapable of correlating new patterns of alerts due to knowledge or rules limitations in terms of updating difficulty and beyond expert's experience. Therefore, a hybrid-based AC (HAC) model is proposed for offering multiple types of correlations among known and new alerts that can deal with the mention.

CHAPTER THREE
METHODOLOGY

3.1 Introduction

This chapter presents the methodology on designing and investigating the architecture of the proposed Hybrid Alert Correlation (HAC). It starts with the addressed problem situations and their applied solution concepts. Based on this, the design and research framework is provided. The rest of the chapter provides detailed description of on how to achieve the objectives, performance measurements, the dataset used in this research as well as experimentation tools and methodological assumptions that are adopted in the development of HAC.

3.2 Research Design

This research addresses the issues of improving the quality of alerts that are generated by multiple NIDSs and recognizing the attack strategy from the unrelated alerts. It is executed through a series of experiments and testing to achieve the goal of objectives of the research. This approach is preferred as the main method due to certain characteristics, such as performance measures, dataset evaluations and the usability of the results. The method on experiments and testing are conducted based on the identified problems under these issues and the coverage of each objective in this research as represented in figure 3 below. The solutions for each problem are designed based on the following five concepts:

i. Alert Preprocessing: The receiving multi-sensors alerts are unstructured and unformatted. Moreover, the alerts are represented by non-numerical values and un-scaled. Such raw alerts are unacceptable for automated analysis. Thus, the solution is to standardize the alerts into a unified form and normalize it into an acceptable input

ii. Feature selection method: Network traffic has many features to measure. The problem is that with the huge amount of network traffic we can measure many irrelevant features. These irrelevant features usually affect the performance of detection rate and consume the IDSs resources.

iii. Alert Clustering: In NIDSs alerts are detected in time sequence. Unfortunately, such sequence cannot reveal the attacks steps launched by the attacker since an attack step may produce single or many alerts. The alerts produced are randomized and dependent to the NIDSs capability. Some NIDSs are very sensitive and may produce high volume of false positives. Others may produce repeated alerts to the same event detected in the network packets. Thus, alerts need to be grouped in order to differentiate false positives, redundancies and true alerts. This can be achieved by clustering.

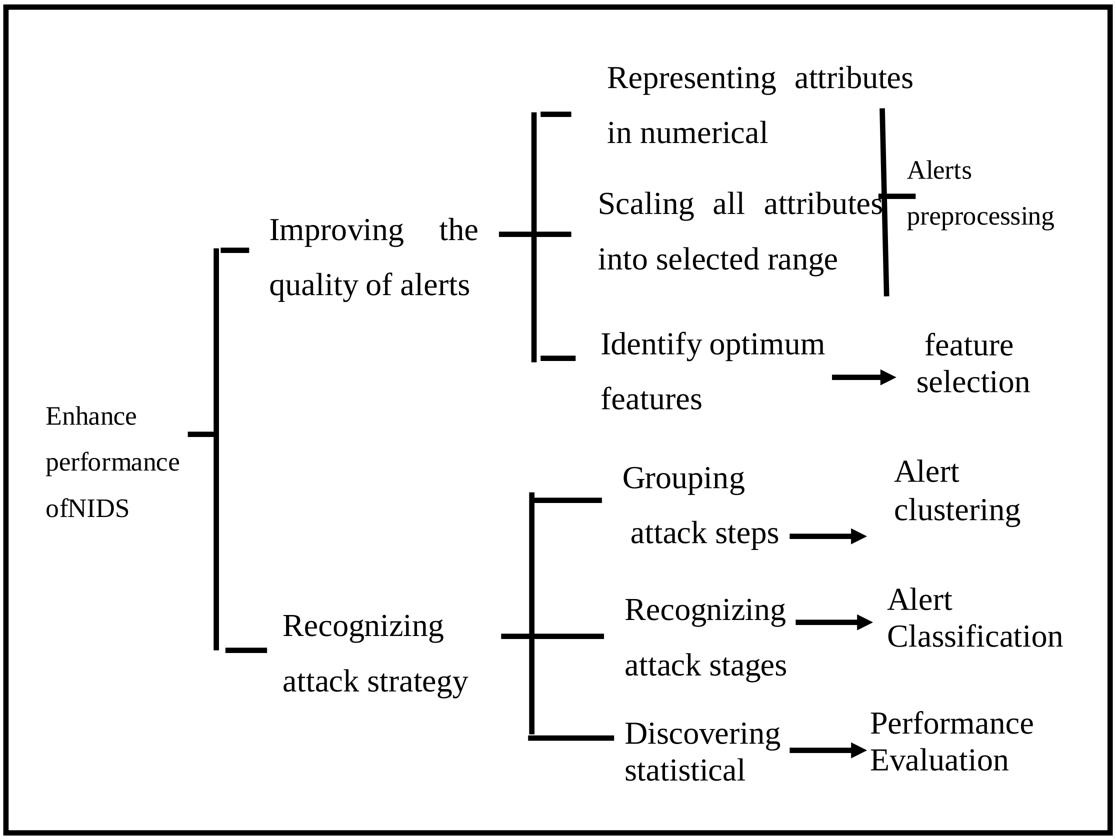

Enhance performance ofNIDS
Improving the quality of alerts
Recognizing attack strategy
Representing attributes in numerical
Scaling all attributes into selected range
Alerts preprocessing
Identify optimum features
feature selection
Grouping attack steps
Alert clustering
Recognizing attack stages
Alert Classification
Discovering statistical
Performance Evaluation

Figure 4: Mapping of problems and solutions

iv. Alert Classification: Although the alert clustering can successfully reveal the attack steps, it cannot recognize the attack stages. For well-known alerts, the causes usually given in the NIDS's signature files or/and manually labeled by the SA based on their professional technical knowledge and experiences. But for new alerts, the correlation approach must have learning capability to predict their causes. Prediction of attack stages membership can be performed by classification.

v. Performance evaluation: Alerts need to be correlated not only effectively, but also completely and optimally. Therefore, the correlation performance needs to be improved. To offer optimality, correlation among attributes has to be considered as well because some attributes may have dependency relationships within each other. For example, a particular targeted port number causes certain IP address to be spoofed or becomes a stepping stone in an attack path. Such case shows that the port number attribute is dependent to the IP address attribute. Thus, the attributes dependency strength has to be measured.

vi. Correlation Improvement and Attributes Dependency Measurements. Alarms should be correlated to improve the detection rate and execution time. Hence, the correlation productivity should be enhanced to offer maximum, correlation amongst features should be measured as well since some features can have strong connections amongst each other. For instance, a specific port number under attack, activates certain IP address to be spoofed or turn into a stepping stone in an attack route. In such a situation. the port number feature is reliant to the IP address attribute. Hence, the features dependency strength has to be computed.

The mapping on AC problems with the respected solution concepts is used as a guidance to design the research framework as provided in the next section.

3.3 Research Framework

The research framework for developing proposed Hybrid Alert Correlation model (HAC) is illustrated in Figure 4. The input is the synthetic data set like NSL KDD dataset. The output is the proposed HAC model that integrates the optimum feature selection technique, structural AC technique, causal AC technique and statistical-based correlations for complete discovery of alerts relationships. It consists of six tasks:

i. The purpose of data processing is to prepare and preprocess the raw alerts that are generated by multiple NIDSs. In order to perform and support automated analysis or correlation on the alerts. In this research normalization, discretization, splitting and merging activities are included.

ii. In this phase, irrelevant and less important features are removed. An ensemble for feature evaluation and feature selection algorithms were invoked to select the set of relevant features.

iii. Dimension reduction is for reducing the alerts high dimensionality using PCA. It can produce good performances due to the elimination of insignificant information from the alerts.

iv. Structural AC groups the alerts based on the similarities of several attributes values using unsupervised learning algorithm. Series of attack steps are revealed by identifying the number of clusters produced. Common attack steps can be recognized by looking at the large clusters.

v. Post clustering deals with discarding redundant alerts and false positives for improving the alerts quality. It is based on Alert Merging and Fusion, Alert Verification and Prioritization and Alert Filtration to remove the identified redundant and false positive alerts

vi. The purpose of enhanced CAC is to predict the membership of each new alert into the predetermined classes or attack stages. In this work a comprehensive analysis on prediction accuracy of standard classifiers and four different ensemble methods, bagging, boosting, voting and stacking is performed in order to determine the algorithm with high detection accuracy and reduce false positive rate.

vii. The hybrid AC aims to produce better classification accuracy and to offer more complete correlation compared to existing works. The hybridization of Clustering, Post-Clustering and Classification is conducted sequentially in order to recognize the attack strategy and improve the overall correlation performance

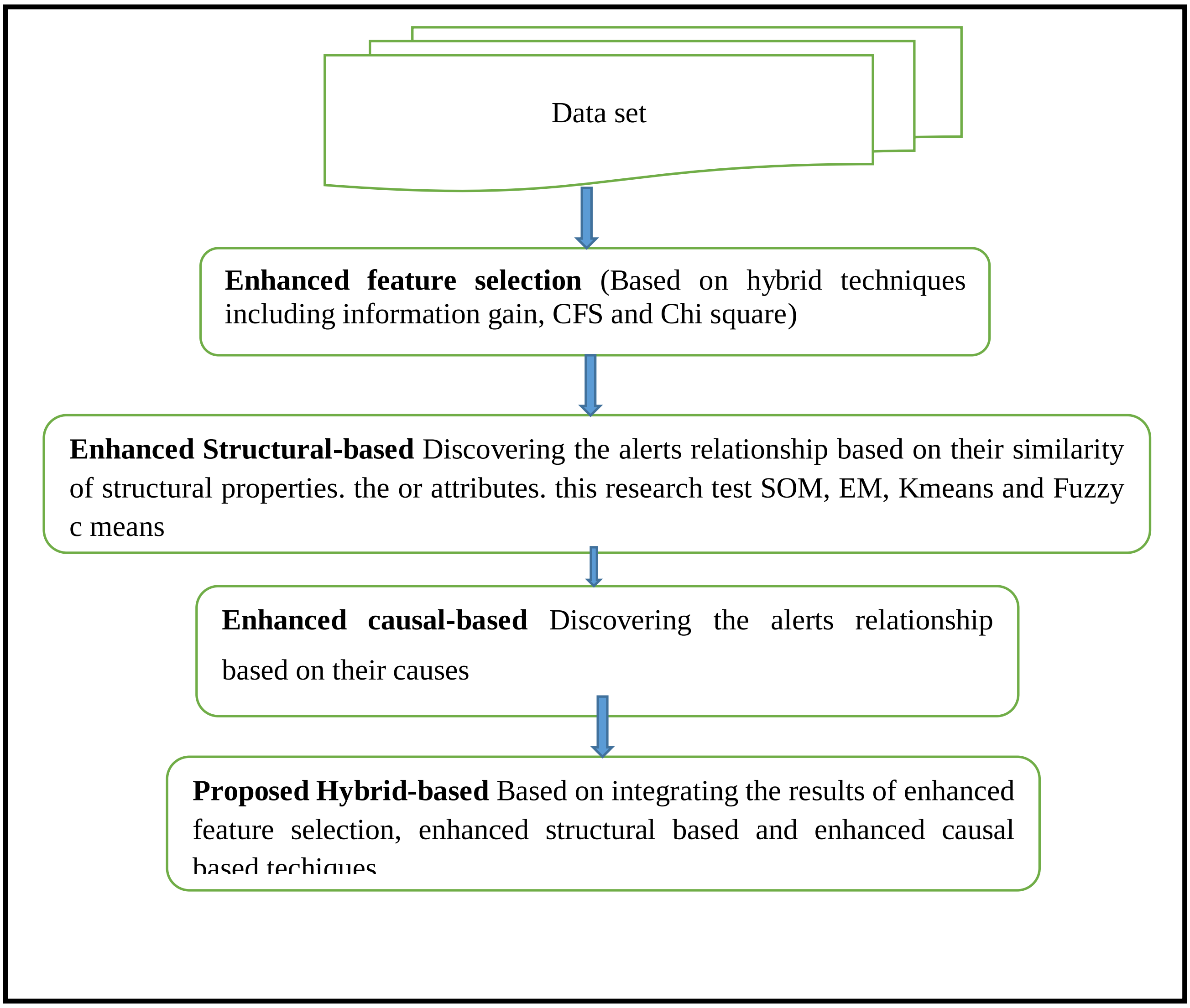

Data set
Enhanced feature selection (Based on hybrid techniques including information gain, CFS and Chi square)
Enhanced Structural-based Discovering the alerts relationship based on their similarity of structural properties. the or attributes. this research test SOM, EM, Kmeans and Fuzzy c means
Enhanced causal-based Discovering the alerts relationship based on their causes
Proposed Hybrid-based Based on integrating the results of enhanced feature selection, enhanced structural based and enhanced causal based techiques

Figure 5: Research framework

The steps used in this investigation are discussed in the subsequent paragraphs

3.3.1 Data Pre Processing

To make efficient use of the available dataset for analysis the data preprocessing is required to provide solutions to Clean the data to remove noise and duplicate information and then deal with any incomplete or missing data an efficient algorithm based on normalization and discretization techniques. The different features are both integers and characters, so in order to apply the algorithm to the data set the features with characters has to be converted into integers while all of the character values has to be changed into a numeric value. The numeric value can then later on be changed into the original value which is a string. This is what is done to features like protocol type, service, flags and so on.

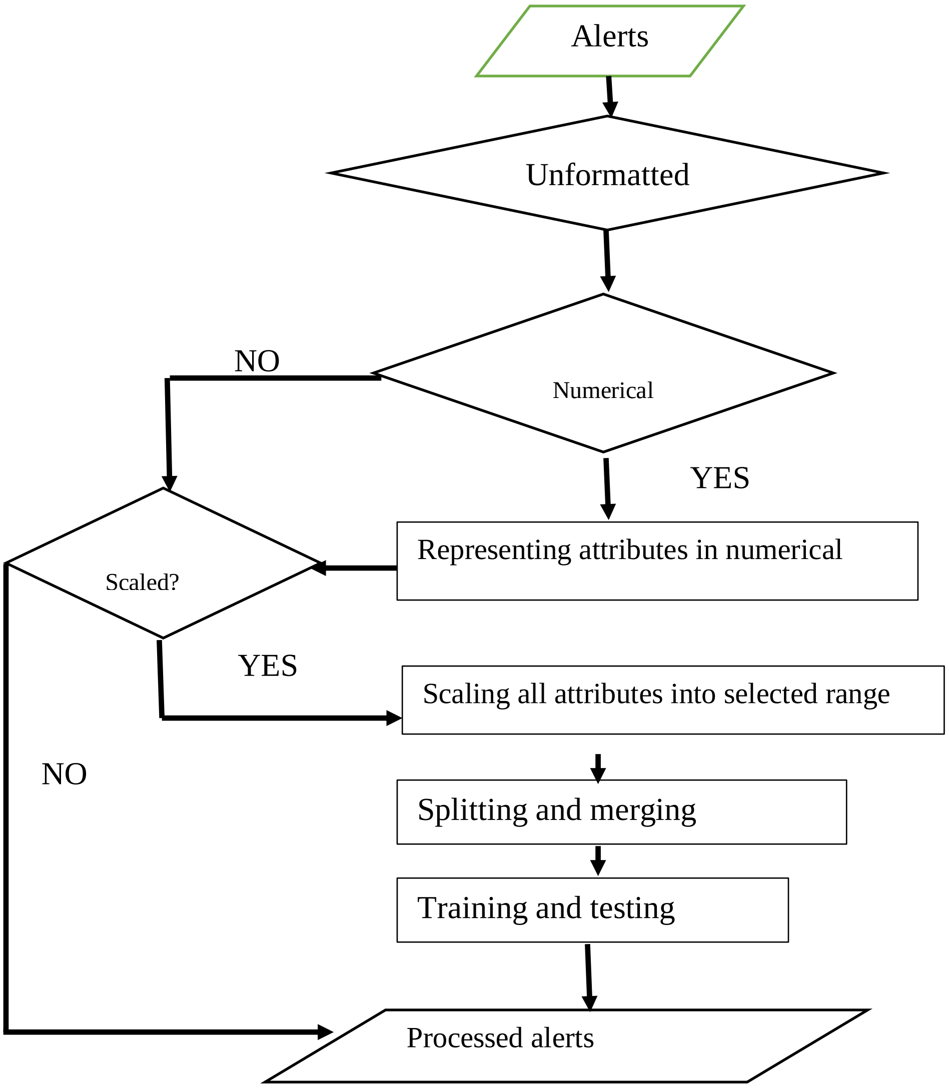


Figure 6: The flowchart of alert preprocessing

1.8.11 3.3.2 Training and Testing Data Set

The machine learning techniques has two phases: a training stage and a testing phase. The training phase set of rules are accountable for developing the profiles that model the training data. The testing phase set of rules use the evidence in the profile to organize connections that are unidentified by the set rules.

Table 2: 10% NSL KDD training dataset preprocessing results

Class	#of instances before	% to all instances	# of instances after	% to all instances	% of reduction
Normal	97,278	19.69%	87,832	60.33%	9.71%
DOS	391,458	79.24%	54,572	37.48%	86.06%
R2L	1,124	0.23%	997	0.68%	11.30%
U2R	54	0.01%	54	0.04%	0.00%
PROBE	4,107	0.83%	2,131	1.46%	48.11%
Total	494,021		145,586		70.53%

Table 2 shows the class distribution and statistics of the reduction of repeated records in the NSL KDD dataset. In this phase, we could remove about 70.5% of redundant and repeated records. This large number of redundant and repeated instances (348,435 instances out of 494,021 instances) causes a major problem while training classifiers, and results in biased classification results. Even after removing these records, NSL KDD dataset still has a major problem that affects the classification results. The problem is the unbalanced and inhomogeneous distribution of attacks and normal instances. There are about (60.33%) of NORMAL class instances, (37.48%) DOS class instances, (1.46%) of PROBE class instances, (0.68%) of R2L class instances, and (0.04%) of U2R class instances. This unbalanced distribution of different classes of NSL KDD dataset biased the classification results to the classes with major instances. This resulted in lower detection performance for classes with low instances, such as U2R and R2L classes. By studying the classification results while using the full 41-features we noticed that relevant of misclassification occurred between attack classes and Normal class. To solve this issue, we created four class-based datasets: (NORMAL + DOS), (NORMAL + PROBE), (NORMAL + R2L), and (NORMAL + U2R). Each of these dataset contains all NORMAL instances plus all instances of only one attack type. The four datasets were used along with the original dataset (NORMAL + all attack type classes) to search for the best set of relevant features. If the training data set is incomplete, it is made complete by replacing the missing values by either one of the method:

1. Replace with mean 2. Replace with median 3. Replace with mode

1.8.12 3.2.3 Data Normalization Technique

Data normalization is a process of scaling the value of each feature into a well-proportioned range, so that the bias in favor of features with greater values is eliminated from the dataset (Ambusaidi et al., 2015). Several techniques are available for normalization and includes Z score, min max normalization and decimal scaling.

A Z-score is a numerical measurement used in statistics of a value's relationship to the mean (average) of a group of values, measured in terms of standard deviations from the mean. If a Z-score is 0, it indicates that the data point's score is identical to the mean score. A Z-score of 1.0 would indicate a value that is one standard deviation from the mean. Z-scores may be positive or negative, with a positive value indicating the score is above the mean and a negative score indicating it is below the mean. Every attribute within each record is scaled by the respective maximum value and falls into the same range of [0-1]. Normalization follows equation 15,

eqn 15

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

where X $_{min}$ is the lowest value for variable X,

X $_{max}$ is the maximum value for variable X.

For a specific symbolic feature, we assigned a discrete integer to each value and then used equation (15) to normalize it.

1.8.13 3.2.4 Discretization Method

Discretization is a process of mapping continuous attributes into nominal attributes. The main objective of the discretization process is to discover a set of cut points, which divide the range into small number of intervals. Every cut-point is a real value within the range of the continuous values, which splits the range into two intervals one is greater than the cut-point and the other is less than or equal to the cut-point value. Data discretization methods is used to condense the amount of values of continuous element by dividing the range of the attribute into small intervals. Interval labels can be used to replace actual attributes value. Replacing values of a continuous attribute by a small number of interval labels reduce and simplifies the original data set. Discretization process is an important preprocessing technique for reducing time of network traffic analysis. Currently, firms are using cooperative network intrusion detection systems to deliver an enhanced detection and comprehensive interpretation of intrusion events. This contributes to the variety of productivity setups. In the process of correlating alerts those diversified formats should be transformed into a unified standard language (Nagle & Chaturvedi, 2013).

This work implements equal interval width and equal frequency intervals Unsupervised Discretization Methods. Equal interval width method divides the range of observed values for a feature into k equal sized bins, where k is a parameter provided by the user while equal frequency intervals requires a feature’s values to be sorted, and assigns 1/k of the values to each bin. Verma, (2016) describe a variation on equal frequency intervals called maximal marginal entropy that iteratively adjusts the boundaries to minimise the entropy at each interval.

Table 3: Codification of NSL KDDTest+ and NSL KDDTrain

Feature Index	Feature Name	Codification
2	protocol_ type	ICMP = 1 ,TCP = 2, UDP = 3
3	Service	s_l = 1 , s_n = 70
4	Flag	F_1 = 1, f_n = 11

42	attack type	DoS = 1, U2R = 2 , R2L = 3
		Probe = 4 , Normal = 5

‘Service’ Feature codified into numbers 1 through 70 with a sequence of ascending name as well as ‘flag’ features from 1 to 11

1.8.14 3.2.5 Splitting and Merging Processes

The research implements a service-based classifiers, hence the initial data sets including training, validation, and testing data first should be separated in a procedure called splitting. The process of splitting, the records are assembled based on the service feature. There are sixty-seven varieties of services that have different numbers of examples. In those groups with minor types, it is not practical to build a classifier for each of them. hence, the researcher, introduces a merging process which pools those minor services into a pseudo-service called majors. The researcher maintains a group of eight principal services which hold more than ten thousand records and combines the other services into a minors' type. Consequently, the research adopts nine subgroups and creates nine profiles. Testing and validation process are split and merged on the defined services.

3.4 Feature Selection Method

Data mining on huge amounts of data is time-consuming operation, making such analysis impractical or infeasible. Data reduction technique have been used to analyse reduced representation of the dataset without compromising the integrity of the original data and yet producing the quality knowledge. As mentioned by Mukosera et al., (2014), NSL KDD dataset contains some challenges that produced defective evaluation results. The main problem is the huge number of duplicates occurrences biased learning algorithm to the groups with many recurring instances. Whereas the less recurrent instances including U2R and R2L are normally more destructive to network will have no effect in learning process. The researcher applied data cleansing and data reduction techniques to resolve this matter. The recurring instances in the '10% NSL KDD' train dataset and the NSL KDD test set were removed, only the non-repeating instances are stored. Existing datasets which contains symbolic data was codified in order to be processed by the system. Training data that have been codified get into the next step, the step of data normalization into the same range of [0-1]. At this step, training data that are not well distributed will be normalized so that the distribution of the data become normal. The goal is to achieve stability of data distribution and useful for adjusting data value with a range of activation functions used in the network.

1.8.15 3.4.1 Ensemble-based Multi-Filter Feature Selection (EMFFS) Method

In this phase, irrelevant and less important features are removed. An ensemble for feature evaluation and feature selection algorithms were invoked to select the set of relevant features.

a novel feature selection model is proposed based on hybridizing feature selection techniques (information gain, correlation feature selection and chi square). The experiment, select attribute set based on the repetition of attribute from four scheme. Existing FS that are employed in experiments are 1) Correlation Feature Selection (CFS) based evaluator with Best-first searching method, 2) Information Gain (IG) based Attributes Evaluator with ranker searching method, and 3) Chi Squared Eval and Ranker searching method we obtained

Table 4: Attribute evaluators and search methods used

Attribute evaluator: correlation-based feature selection (CFS), Information Gain and Chi Square	
Search method	Description
Best first	Searches the space of attribute subsets by greedy hill climbing augmented with a backtracking facility.
Greedy stepwise	Performs a greedy forward or backward search through the space of attribute subsets.
Rank search (info gain)	Evaluates the worth of an attribute by measuring the information gain with respect to the class.

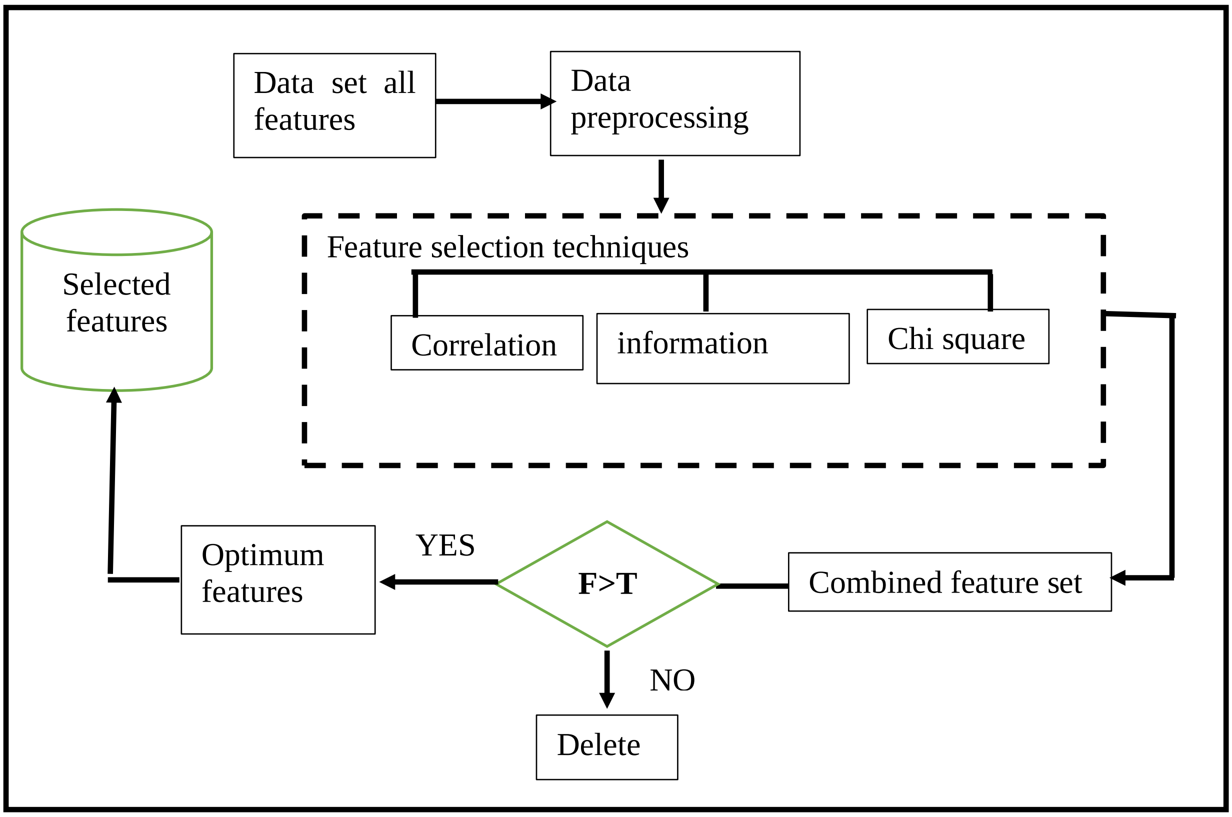


Figure 7: Proposed Ensemble-based Multi-Filter Feature Selection (EMFFS) Method

Each algorithm evaluated each class dependent dataset created resulted in a relevant set of features for each particular class. The researcher considered only features that are selected by ten folds (like., k = 10). On the other hand, features that not selected by any algorithm were irrelevant and removed from the list. Output of this phase is a reduced set of common relevant features that were ranked by its relevance value for each attack class.

1.8.16 3.4.2 Best Features Selection

In this phase, we selected the best set of relevant features. The Features selected in the previous phase were ranked based on their relevance value to each attack class. This phase contains two independent groups: Gradually ADD Feature and Gradually DELETE Feature. The reason is to apply two independent methods to get the optimum features. Two ordered list of characteristics were realized. One for the that contains required characteristics selected by dissimilar procedures. Where the other contains characteristics that are appropriate to overall attack groups. The common characteristic fields at the end class of these two ranked lists are excepted and removed one by one. The rest of features re-evaluated another time to ensure that removing these features has not affect to the whole detection accurateness and performance. The algorithm used in this phase is shown in appendix III

3.5 Enhancing Structural Correlation Based on Unsupervised Learning Techniques

The cyber threats evolve and progress in a drastic way, modern organizations adopts multiple Network Intrusion Detection Systems (NIDSs) to improve discovery and to deliver complete assessment of intrusion events. Nevertheless, NIDSs generate a huge quantity of alerts even in a single day and overwhelms network administrator as they involve a lot of human effort in building the system and maintaining it. Therefore, computerized and intelligent clustering is essential to discover their structural relationship by grouping alarms with similar attributes. The main goal in this stage is to enhance the structural based alert correlation model to improve the quality of alerts and detection capability by grouping alerts with common attributes based on unsupervised learning techniques. Our focus is to minimize human intervention as much as possible, but not to replace them. Therefore, an unsupervised learning-based clustering model is proposed to reduce the number of alerts and to discover the attack steps launched by attackers.

1.8.17 3.5.1 Alert clustering Techniques

The goal of this phase is to find the best integration of PCA and unsupervised learning algorithm for clustering intrusion alerts. This section compares four unsupervised algorithms which includes Self-organizing maps (SOM), K-means, and Fuzzy c-means (FCM) and Expectation and maximization (EM) technique to find which algorithm will be able to offer more detection accuracy. The output is a network anomaly detection method based on Principal Component Analysis (PCA) and unsupervised learning model that gives optimum results to aggregate similar alerts and to reduce the number of alerts.

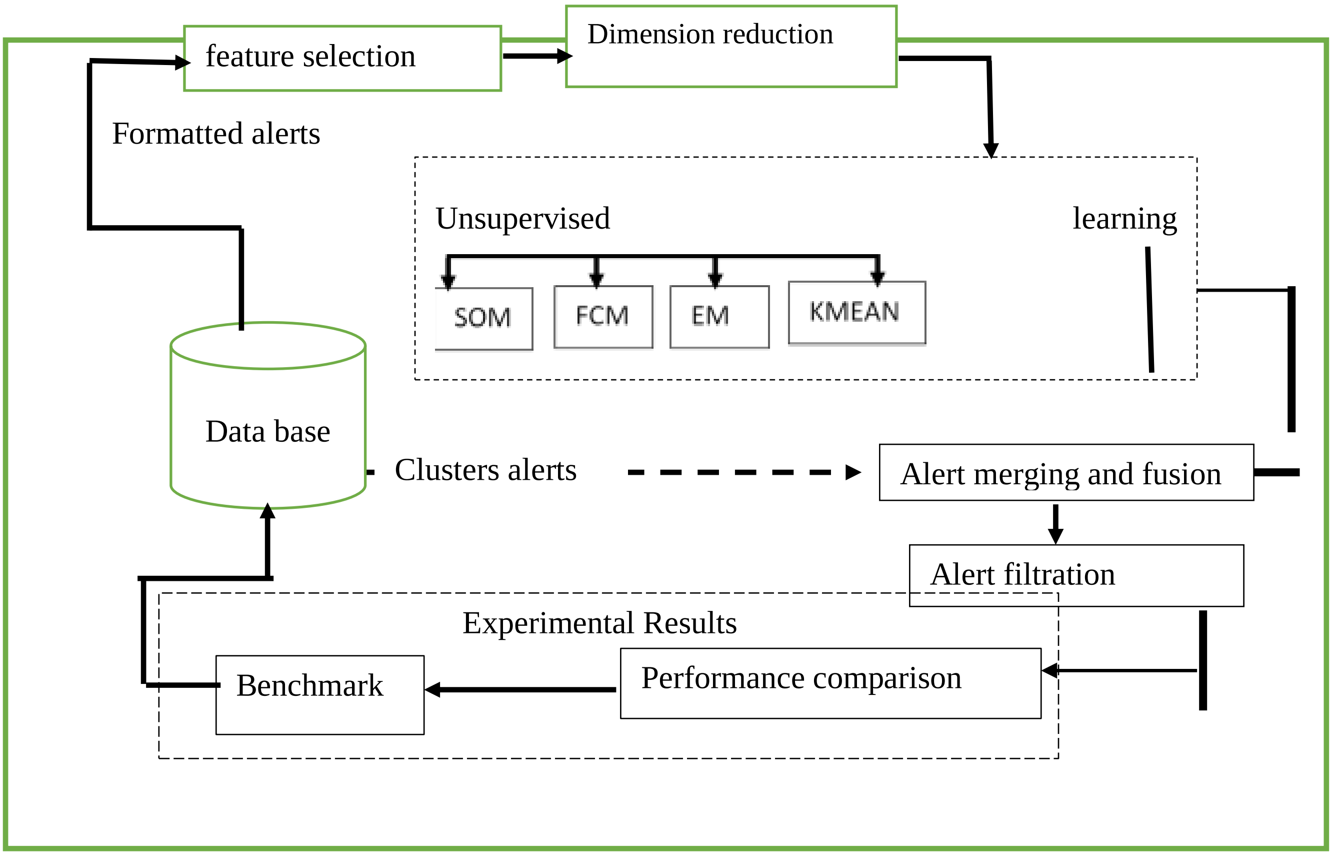


Figure 8: The flowchart of enhanced structural-based alert correlation model

Steps

Step (1) Read the pre-processed alerts as inputs.

Alerts that have been processed by Multi-Filter Feature Selection (EMFFS) Method are read from the database as inputs clustering phase.

Step (2) Reduce the alerts high dimensionality.

All alerts with their attributes are dimensionally reduced using statistical PCA

Step (3) Adopt unsupervised learning algorithm which gives the highest accuracy. Expectation Maximization (EM), (K-means, FCM and SOM. unsupervised learning algorithm are tested and compared.

Step (4) Verify and prioritize the alerts. All alerts are automatically cross-checked with the NIDS's signature files to verify false positives and invalid alerts.

Based on information in the signature files as well, the alerts are also ranked into low, medium and high severity level to identify the risks of each alert.

Step (5) Filter out the low quality alerts. The low quality alerts (redundant, false positives and invalid alerts) are deleted to improve the quality of alerts and to reduce the number of alerts.

Step (6) Measure and validate the clustering and post-clustering performances. The performances of the proposed clustering system can be measured using predefined measurements.

Step (7) Save the analysis and experimental results. The analysis and experimental results are recorded and saved in the database. It includes the details on all of the identified clusters attack steps as well as the statistical analysis.

1.8.18 3.5.2 Unsupervised Learning Parameters

For comparison purpose, the Self-Organizing Maps (SOM), Fuzzy c-means (FCM) and K-means and Expectation and maximization are experimented. To train the alerts in unsupervised learning, the initial number of clusters must be provided. The number is randomly varied from 1 to 50 clusters for FCM, K means, and EM in order to find the optimal results. The SOM has different parameters to tune. They are topology, distance function, learning rate and epochs:

a. The topology of the neural network (or lattice type) used is hexagonal since it is best suited for visual display. There are three types of lattice configuration tested: 4x6, 5x7, and 6x8.
b. The LINKDIST distance function is set to determine the distance between points in the map.
c. The learning rate was set to 0.4 to speed up the network convergence to the desired state.
d. The maximum number of training steps (epochs or iteration) is tuned at 1000. For testing, the epochs are varied from 100 to 5000 (with interval 100 epochs). For each dataset, 70% of them were used for training, 30% for validation and testing.

3.6 Enhanced Causal-based Alert Correlation Model.

One of the major developments in machine learning in the past decade is the ensemble method, which generates highly accurate classifier by combining several classifiers generated from one or more learning algorithms (Ikram & Cherukuri, 2016; Perez, Astor, Abreu, & Scalise, 2017). In this work a comprehensive analysis on prediction accuracy of standard classifiers and four different ensemble methods, bagging, boosting, voting and stacking is performed in order to determine the algorithm with high detection accuracy and reduce false positive rate. Three different experiments on NSL KDD data set are conducted and their performance compared and evaluated based on accuracy, false alarms and computation time.

The steps include.

Step (1) Read alerts from database as inputs.

Step (2) Reduce the alerts high dimensionality.

Step (3) Design the classifier

Step (4) Adopt Ensemble supervised learning algorithm for the classifier.

Step (5) Measure and validate the classification performance.

Step (6) Save the analysis and experimental results.

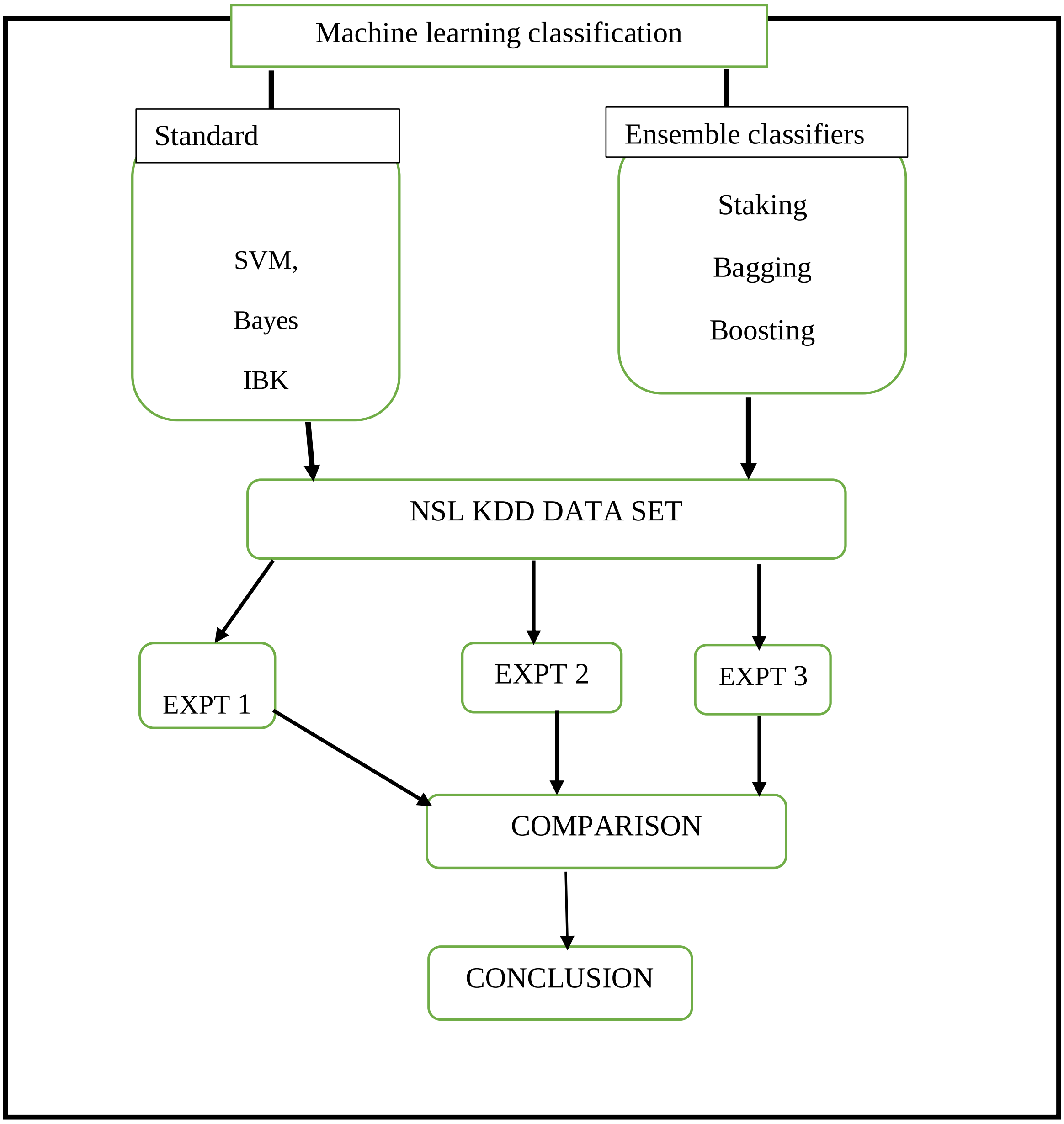


Figure 9: performance Comparison of standard classifiers and ensemble classifiers

1.8.19 3.6.1 Application of Bagging and Boosting with Five Classifiers

The experiment is conducted with two ensemble learning techniques, bagging and boosting and five classifier using 10-fold cross validation. The single classifier includes Bayes Net, IBK, ANN, Jrip and SVM. The conducted experiments are evaluated according to four performance measures which are accuracy, precision, recall, and f-measure.

Table 5: Bagging and Boosting Parameters and their meaning

Parameter	Description
Bag size percentage	**Size of each bag as a percentage of the training set size**
Classifier	The base classifiers to be used
Min iterations	The number of iterations to be performed
Seed	The random number seed to be used
Weight threshold	Weigh threshold for weight pruning

3.6.2 Application of stacking as a multi classifier with five classifiers

In the stacking approach, we compare five different algorithms and SVM as a base learner and stacking as a multi classifier learner are used. This research employs a combination of Bayes Net, IBK, ANN, J48 and JRip. The classifications anticipated from the base learners is taken as input variables into a stacking model learner. Every input classification algorithm calculates anticipated classifications results using tenfold cross validation technique after which complete performance characteristic is calculated. Hence the stacking model learner attempt to learn from the information on how to conglomerate the estimates from the dissimilar models to attain optimum classification accurateness.

Support Vector Machine, (SVM) is a machine learning algorithm implemented for classification, regression and outlier detection. It is one of the applicable correct and strong procedures for classification and widely used in intrusion detection system to deliver optimum security and takes minimum time to discover attacks (Chand et al., 2017; Khorshid et al., 2015). The major features of SVM according to Thaseen & Kumar, (2016) include: Deals with very large data sets efficiently, Multiclass classification can be done with any number of class labels, High dimensional data in both sparse and dense formats are supported, Expensive computing not required and can be applied in many applications like e-commerce, text classification, bioinformatics, banking and other areas. Even though SVMs are limited to making binary classifications, their superior properties of fast training, scalability and generalization capability give them an advantage in the intrusion detection application. Finding cost-efficient ways to speed up or parallelize the multiple runs of SVMs to make multi-class identification is also under investigation (Hussain, Lalmuanawma, & Chhakchhuak, 2015).

Table 6: parameters description

Classifiers	The base classifiers to be used (Bayes Net, IBK, ANN, J48 and JRip).
debug	If set to true, classifier may output additional information to the console.
metaClassifier	The meta classifiers to be used in our case SVM.
numFolds	The number of folds used for cross-validation. The value to use is 10
seed	The random number seed to be used

1.8.20 3.6.3 Combination of Five Distinct Classifiers Using Voting Technique.

The classification module of hybrid technique is composed of combination of five distinct classifiers based on Voting Technique. This experiment is designed on the idea that each IDS is efficient in detecting a specific attack type. The techniques include Bayes Net, SVM, IBK, J48 and random forest with voting as a multi classifier.

classifiers -- the base classifiers to be used (bayes net, svm, ibk, j48 and random forest**).**

combination rule -- the combination rule used. in our case majority voting

debug -- if set to true, classifier may output additional info to the console.

seed -- the random number seed to be used.

3.7 The Proposed Hybrid Machine Learning Model

Anomaly-based IDSs recognize the anomalous, unfamiliar activities on a network and label them as attacks and it doesn't require some explicit knowledge. The major drawback with this technique is that it produces a lot of false alarms, unable to identify known and new attacks. The proposed system solves this problem by integrating several classification algorithms. It comprises of a Multiple IDS Unit (MIU) which contains five IDS units resulting from five dissimilar systems. This section presents an intrusion detection system that combines heterogeneous anomaly detection techniques for network attack detection. The main idea is that every IDS is effective in discovering an exact kind of attack. The proposed fusion model employs a multi-level processing architecture, which takes into account the categories of techniques and algorithms used for detection. The five processing levels includes.

a) Feature selection Extracts the optimum features from synthetic dataset based on ensemble feature selection methods
b) Dimension Reduction uses PCA to reduce the dimensionality of the alerts for optimal correlation performance.
c) Unsupervised Learning Algorithm clusters alerts into groups/attack steps to discover the structural correlation among the alerts.
d) Post-Clustering Algorithms improve the quality of alerts by filtering out the unwanted low quality alerts (redundant, false positives and low-risk alerts).
e) Ensemble Supervised Learning Algorithm classifies alerts into classes/attack stages to discover the causal correlation among the alerts.
f) Statistical Correlation Tests calculate the strength of dependencies among the alerts attributes to discover the statistical correlation,
g) Benchmark evaluates and compares with current works

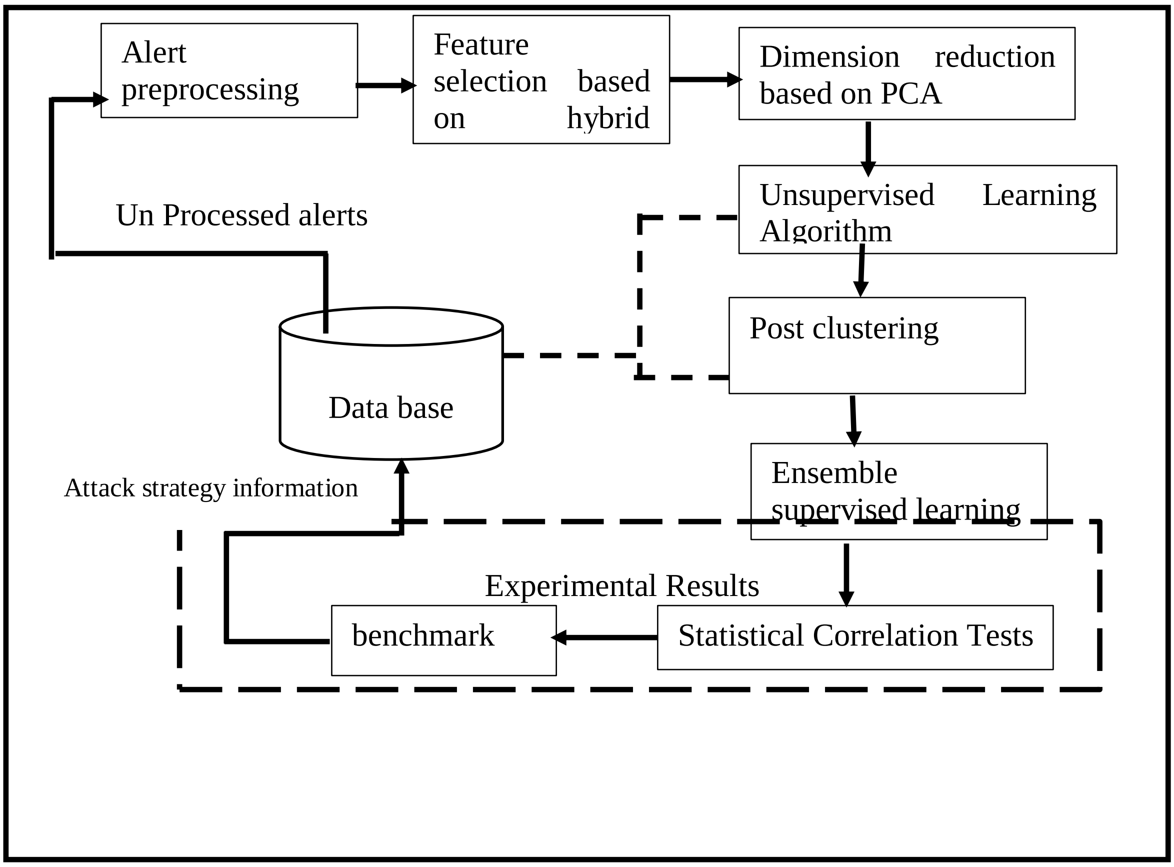

Alert preprocessing
Feature selection based on hybrid
Dimension reduction based on PCA
Unsupervised Learning Algorithm
Un Processed alerts
Post clustering
Data base
Ensemble supervised learning
Attack strategy information
Experimental Results
benchmark
Statistical Correlation Tests

Figure 10: Proposed Hybrid Intrusion Detection System

3.8 Description of the Network Traffc Data Sets

1.8.21 3.8.1 The UNSW-NB15 Dataset

The UNSW-NB15 dataset was published in 2015 by Moustafa at el, (2014) for research purposes in IDS. It is a hybrid of attack activities include real traffic and synthesized activities in a computer network traffics and comprises of nine different moderns attack types as compared to fourteen (14) attack types in KDD'99 datasets activities of normal traffic that were captured with the change over time (Janarthanan, 2017). The UNSW-NB15 dataset has forty-nine (49) features that comprised the flow based between hosts (like., client-to-server or server-to-client) and the packet header which covers in-depth characteristics of the network traffic. This data set contains 2, 540,044 observations

In UNSW-NB15 data set, there are nine categories of attacks:

1. Fuzzers: In this attack, randomly generated data is feed into a suspend program or network.

2. Reconnaissance: Attacker gathers information from the system and stimulates the attacks.
3. Shellcode: It is code used as the payload of a network packet to exploit network attacks.
4. Analysis: This attack includes port scan, spam and HTML files penetrations.
5. Backdoors: Access of a system is gained by silently bypassing the security mechanism.
6. Denial of Service where attempts are to shut down, suspend services of a network resource remotely making it unavailable to its intended users by overloading the server with too many requests to be handled.
7. Exploits: The attacker exploits the vulnerabilities of the system through the known loopholes of the system.
8. Generic: The attack is implemented without knowing how the cryptographic primitive is implemented and works for all block ciphers.
9. Worms: The attack replicates itself to spread through the network.

Table 7: Attack Distribution in UNSW-NB15 Data Set

Category	**Training**	**Set Testing set**
Normal	**56,000**	**37,000**
Analysis	**2,000**	**677**
Backdoor	**1,746**	**583**
DoS	**12,264**	**4089**
Exploits	**33,393**	**11,132**
fuzzers	**18,184**	**6062,**
Generic	**40,000**	**18,871**
Reconnaissance	**10,491**	**3,496**
Shellcode	**1,133**	**378**
Worms	**130**	**44**
Total Records	**175,341**	**82,332**

The UNSW-NB15 dataset is divided into two Training datasets (82, 332 records) and a Testing dataset (175, 341 records) including all attack types and normal traffic records. Both the Training and Testing datasets have 45 features. The features scrip, sport, dstip, stime and ltime are missing in the Training and Testing dataset.

The UNSW-NB15 data set has several advantages when compared to the NSLKDD data set (Datasets, Mogal, Ghungrad, & Bhusare, 2017). First, it contains real modern normal behaviors and contemporary synthesized attack activities. Second, the probability distribution of the training and testing sets are similar. Third, it involves a set of features from the payload and header of packets to reflect the network packets efficiently. Finally, the complexity of evaluating the UNSWNB15 on existing classification systems showed that this data set has complex patterns. This means that the data set can be used to evaluate the existing and novel classification methods in an effective and reliable manner.

1.8.22 3.8.2 The NSL-KDD Dataset

In this study, the researcher uses the NSL-KDD dataset which is derived from original KDD-99 and has eliminated some of its drawbacks. It has the following characteristics (Manandhar, 2014):

i. The existing KDD99 dataset contain huge numbers of duplicated records which makes the learning algorithms to be inclined towards recurrent records like probe and DOS hence preventing them from detecting unknown records that are in less frequent attack group like U2R and R2L attack categories that an attacker may take advantage of to infiltrate a computer networks (Panda et al., 2015).

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

iii. The numbers of records in the train and test sets are reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and comparable.

The simulated attacks in the NSL-KDD dataset can be grouped into four categories (Jain & Rana, 2016; Parsaei et al., 2016; Shahadat *et al.*, 2017).

i. Denial of service attack (Dos), where attempts are to shut down, suspend services of a network resource remotely making it unavailable to its intended users by overloading the server with too many requests to be handled. e.g. syn flooding. Relevant features include source bytes and percentage of packets with errors. Examples of attacks includes back, land, Neptune, pod, Smurf, teardrop

ii. Probe attacks, where the hacker scans the network of computers or DNS server to find valid IP, active ports, host operating system and known vulnerabilities with the aim discover useful information. Relevant features include duration of connection and source bytes. Examples includes IP sweep, n map, port sweep, Satan

iii. Remote-to-Local (R2L) attacks, where an attacker who does not have an account with the machine tries to gain local access to unauthorized information through sending packets to the victim machine in filtrates files from the machine or modifies in transit to the machine. Relevant features include number of file creations and number of shell prompts invoked. Attacks in this category includes ftp_ write, guess_ passwd, I map, multi hop, phf, spy, warezclient, warezmaster

iv. User-to-Root (U2R) attacks, where an attacker gains root access to the system using his normal user account to exploit vulnerabilities. Relevant features include Network level features – duration of connection and service requested and host level features - number of failed login attempts. Attacks includes buffer overflow, load module, Perl, rootkit

In the KDD’99 data set, each record contains forty-two 42 features (together with its class tag) that hold information about the period. These features are grouped into four classes: basic, content, traffic, and class.

i. Basic attributes. that represent an alert and they are in IDMEF format. Examples of these attributes include timestamp, signature identifier, messages associated with alerts, protocol, IP source and IP destination addresses, source port and destination address, Time to live and identification field.

ii. Content features: The features of suspicious behavior in the data portion should be captured in order to detect attacks. E.g. number of failed login attempts. Those features are called content features. The R2L and U2R attacks normally don’t appear in intrusion frequent sequential patterns, as they have been embedded in the data portions of packets and only request a single connection. While the DoS and Probing attacks involve many connections to hosts and show the attribute of intrusion frequent sequential patterns.

iii. Time-based traffic features: Only the connections in the past two seconds are examined, which have the same destination host/service as the current connection, and of which the statistics related to protocol behavior, service, etc. are calculated.

iv. Connection-based traffic features: Some slow probing attacks scan the hosts/service at an internal much longer than two seconds, e.g. once in every minute, which cannot be detected by the time-based traffic features, as it only examines the connections in the past 2 seconds. In such case, the features of same destination host/service connections can be re-calculated at an interval of every 100 connections rather than a time window

A description of each of the 42 features is listed in the appendices 1

NSL-KDD dataset consists of KDD Test for data testing and KDD Train for data training. Each dataset has 41 features to recognize four types of attacks (DoS, U2R, R2L, Probe) and 1 normal state that occurs on a computer network. The features of the dataset include basic features of TCP connection, content features from domain knowledge and traffic features. This work will apply 5% KDD Train+ to get the proper cluster (6300 data), then 100% KDD Train+ (22544 data) for training data and 100% KDD Test+ (125973 data) for testing data. Dataset composed of normal data and 4 categories of attacks, namely DoS, U2R, R2L and Probe.

Table 8: shows the composition of the training and testing datasets.

Data	**Number of each category**				
	Normal	DOS	U2R	R2L	Probe
NSL KDD Train	67343	45927	52	995	11656
NSL KDD Test	9711	7458	200	2754	2421

The proposed model hybridizes data preprocessing technique, feature selection technique, structural based technique, causal based and statistical correlation investigations to improve the complete relationship performance and evaluate the reliance strength amongst alert features**.**

3.9 Waikato Environment for Knowledge Analysis (WEKA)

Several data mining techniques which includes data cleaning and pre-processing, clustering, classification, regression, visualization and feature selection have been implemented in WEKA (Waikato Environment for Knowledge Analysis) (Revathi, 2000). WEKA also offers some functionality that other tools do not, such as the ability to run up to six classifiers on all datasets, handling multi-class datasets which other tools continue to struggle with tools. WEKA has tools for various data mining tasks. WEKA is considered as a landmark of data mining and machine learning as compared to other data mining and knowledge discovery tools and software like Tanagra, the Konstanz Information Miner (KNIME), and Orange Canvas (Kovac, 2012; Kumar, 2016). Due to its Graphical User Interface (GUI) and easy access it has achieved a wide acceptance in every field. WEKA contains classes which can be accessed by other classes of WEKA. The relevant classes in WEKA are attribute and instance. An attribute is represented by an object of class attributes which contains attribute types, name, type, nominal values of attributes. It is user friendly with a graphical interface that allows for quick set up and operation. WEKA operates on the predication that the user data is available as a flat file or relation, this means that each data object is described by a fixed number of attributes that usually are of a specific type, normal alpha-numeric or numeric values.

Table 9: description of explorer user interface in WEKA

Data Mining Task	Description	Examples
Data Pre-Processing	Preparing a dataset for analysis	Discretizing, Nominal to Binary
Classification	Given a labeled set of observations, learn to predict labels for new observations	Bayes Net, KNN, Decision Tree, Neural Networks, Perceptron, SVM
Regression	Learn to predict numeric values for observations	Linear Regression, Isotonic Regression
Clustering	Identify groups (like., clusters) of similar observations	K-Means, EM,
Association rule mining	Discovering relationships between variables	Apriori Algorithm, Predictive Accuracy
Feature Selection	Find attributes of observations important for prediction	Cfs Subset Evaluation, Info Gain
Visualization	Visually represent data mining results	Cluster assignments, ROC curves

WEKA consists of several user interfaces. But the functionality can be performed by any one of them as they give the same result. In WEKA user interface is classified into four categories

i. **Explorer** – GUI, very popular interface for batch data processing; tab based interface to algorithms. Each of the packages includes Filters, Classifiers, Clusters, Associations, and Attribute Selection is represented in the Explorer along with a Visualization tool which allows datasets and the predictions of Classifiers and Clusters to be visualized in two dimensions.

ii. **Knowledge flow** – GUI where users lay out and connect widgets representing WEKA components. Allows incremental processing of data. WEKA components are selected from a tool bar, positioned a layout canvas, and connected into a directed graph to model a complete system that processes and analyzes data. Components available in the Knowledge Flow: data source, filters, Clusters, classifiers, Evaluation and Visualization

iii. **Experimenter** – GUI allowing large scale comparison of predictive performances of learning algorithms. The experimenter, which can be run from both the command line and a GUI, is a tool that allows you to perform more than one experiment at a time, maybe applying different techniques to a dataset, or the same technique repeatedly with different parameters. For example, the user can create an experiment that runs several schemes against a series of datasets and then analyse the results to determine if one of the schemes is (statistically) better than the other schemes.

iv. **Command Line Interface (CLI)** – Provides a simple command line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface.

3.10 Performance Evaluation Metrics

In cross validation, the available dataset is randomly subdivided into 10 equal disjoint subsets and one of them is used as the test set and the remaining sets are used for building the classifier (Moustafa & Slay, 2015). In this process, the test subset is used to calculate the output accuracy while the N_1 subset is used as a test subset and to find the accuracy for each subset. The process is repeated until each subset is able to assess the set once and to compute the output accuracy of each subset. The final accuracy of the system is computed based on the accuracy of the entire 10 disjoint subsets. The confusion matrix was used to evaluate the performance of the IDS.

A confusion matrix is a specific table layout which allows visualization of the performance of intrusion detection system. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The name stems from the fact that it makes it easy to see if the system is confusing two classes (like. commonly mislabeling one as another). In the binary class IDS, the intrusion detection system is mainly discriminate between to classes, "Attack" class (malicious threats or abnormal data) and "Normal" class (normal data)

Table 10: Confusion Matrix

	Predicted	
	Normal	Attack
Actual normal	TP	FP
Actual attack	FN	TN

In Table 10, The components from the confusion metrics are CM= {TP, TN, FP, FN}, where TP (True positive) is the value of the correctly classified attacks, TN (True Negative) is the number of the correctly classified normal rows, FP (False Positive) is the number of the misclassified attacks, and FN (False Negative) refers to the number of the misclassified normal records.

The selected measurements used for performance validation and benchmark in this research for the enhanced SAC, enhanced CAC and proposed HAC are justified and described in the following:

a. Structural-based Alert Correlation (SAC). The applied measurements for validating the enhanced SAC include:

   i. Clustering Error (CE) are the total amount of alerts which are incorrectly grouped.

   ii. Error Rate (ER) is the percentage of incorrectly grouped alarms, ER = (CE ÷ Entire values of alarms observed) x 100,

   iii. Accuracy Rate (AR) are the percentage of warnings which are correctly grouped can be represented as, AR = 100 – ER, and

   iv. Time is the procedure execution time expressed in seconds.

b. Causal-based Alert Correlation (CAC). CAC model is concerned about how good it can classify the known and new alerts. Therefore, this research implemented the standard classification measurements in validating and evaluating the enhanced CAC model. They are:

   i. TPR: TP/(TP+FN), also known as detection rate (DR) or sensitivity or recall.

   ii. The False Alarm Rate (FAR) is the rate of the misclassified to classified records, as denoted in Equation (16). Equations (16) and (17) allow calculation of the False Positive Rate (FPR) and the False Negative Rate (FNR), respectively. FP/(TN+FP) also known as the false alarm rate.

i. **FPR = FP/ (FP +TN)** Eqn 16

ii. **FNR = FN/ (FN +TP)** Eqn 17

iii. Precision (P): TP/(TP+FP) is defined as the proportion of the true positives against all the positive results.

iv. Total Accuracy (TA): (TP+TN)/(TP+TN+FP+FN) is the proportion of true results (both true positives and true negatives) in the population.

v. F-measure: 2PR/(P+R) is the harmonic mean of precision and recall.

For the proposed correlation models, the optimal setting of the parameters is done based on repeated trials. But, for retesting the current works, the parameters are set based on their information given in their published research resources or papers. In the case that information is not given, the default parameters are adopted. Almost all types of computers can be used to code and run the proposed correlation models because there is no specific special hardware is needed. Furthermore, all the software and tools needed are either freely available or easily purchasable. Nevertheless, the minimum computer hardware requirements are Core-i5 processor, 2.5GHz speed, and 8GB RAM. But, a higher specification is better for maximum installation and smooth experiments.

3.11 Summary

This chapter provides detail research methodology which includes the operational framework, flowcharts, plans and applied performance measurements for the enhanced SAC, enhanced CAC and proposed HAC models. The framework hybridizes three types of correlations; structural, causal and statistical for complete and optimal analysis of multiple NIDSs alerts. In the enhanced SAC, alerts are clustered using EM unsupervised learning algorithm to identify the list of attack steps whereas in enhanced CAC, the alerts are classified to recognize the attack stages memberships. Both SAC and CAC are enhanced based on current works. Their performances are optimized with the application of PCA for dimension reduction in the searching of selecting only principal and significant information are considered during correlations.

CHAPTER FOUR

DATA ANALYSIS, PRESENTATION AND DISCUSSION

4.1 Introduction

This chapter presents the findings, interpretations and discussion of the research objectives as stated in chapter One. The experimental setup, results and analysis are presented in tabular form.

The major objective of the research is to develop a hybrid machine learning model to improve the performance of network intrusion detection system. To achieve this, the research is conducted based on four main phases. They include: to identify relevant feature set based on hybrid feature selection techniques, to enhance structural correlation based on unsupervised machine learning techniques, to enhance causal alert correlation model using supervised machine learning techniques and develop alert correlation model based on hybrid machine learning techniques to enhance the performance of Network Intrusion Detection System.

4.2 Enhanced Feature Selection Based on Multi-Filter Feature Selection (EMFFS) Method

In this objective, a novel feature selection model is proposed based on hybridizing feature selection techniques (information gain, correlation feature selection and chi square). The experiment, select attribute set based on the repetition of attribute from four scheme. Existing FS that are employed in experiments are Correlation Feature Selection (CFS) based evaluator with Best-first searching method, Gain Ratio (GR) Attributes based Evaluator with Ranker searching method, Information Gain (IG) based Attributes Evaluator with ranker searching method, and Chi Squared Eval and Ranker searching method we obtained. In the proposed model, these algorithms select the optimal features group from entire attack categories in NSL KDD dataset (DOS, PROBE, R2L, U2R, and NORMAL class) as indicated from Table 11 to table 14. Table 15 represents the best set of features in each category.

Table 11: The features that distinguish between usual network traffic and the attacks

Feature Selection Techniques	No Of Features	Selected Features
Original Dataset	41	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,31,33,34,35, 36,37,38,39,40,41
Information Gain, Ranker	10	2, 40,3,41,27,26,30,31,32,35
CFS , Best First,	8	2,3,9,23,26,27,34,41
Gain Ratio And Ranker	9	9,23,41,22,36,3,27,35,2
CHI Squared Eval + Ranker	9	2,40,3,41,26,27,30,31,32
Proposed	11	2,3,4,26,27,36,39,41

Table 10 above contain 41features of normal network traffic. After applying hybrid feature selection techniques are reduced to 11 optimum features. The features include protocol type, Service, Flag, serror_rate, rerror_rate, dst_host_diff_srv_rate, Dst_host_srv_rerror_rate,

dst_host_serror_rate

Table 12: The most important features to distinguish between normal network traffic and DoS attacks

Feature selection techniques	**No of features**	**Selected Features**
Original Dataset	41	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,31,33,34,35,36,37,38,39,40,41
Information gain, ranker	10	2, 40,3,41,27,26,30,31,32,35
CFS , best first,	8	2,3,9,23,26,27,34,41
Gain ratio and ranker	10	9,23,41,22,36,3,27,35,2,26
Chi Squared Eval + Ranker	10	2,40,3,41,26,27,30,31,32,20
Optimal Feature	7	2,3,9,26,41.4,27

From table 12, the optimum features to distinguish normal traffic with DOS are protocol type, Service, Flag, Urgent, serror_rate, rerror_rate, dst_host_srv_serror_rate

Table 13: The most important features to distinguish between normal network traffic and Probing Attacks

Feature selection techniques	No of features	Selected Features
Original Dataset	41	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,31,33,34,35, 36,37,38,39,40,41
Information gain, ranker	10	2, 40,3,30,,34,9,33,32,31,38
CFS , best first,	9	2,3,9,24,26,30,34,38,40
Gain ratio and ranker	10	25,9,24,3,2,41,38,40,34,26,
Chi Squared Eval + Ranker	10	2,40,3,33,34,30,32,38,31,37
Optimal Features	7	2,3,9,30,34,38,40

From table 13, the optimum features to distinguish normal traffic with Probe attack are protocol type, Service, Urgent, srv_serror_rate, dst_host_srv_count, Dst_host_ srv_diff_host_rate, dst_host_serror_rate

Table 14: The features that distinguish between the network traffic and Remote to Local (R2L) attacks

Feature selection techniques	**No of features**	**Selected Features**
Original Dataset	41	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,31,33,34,35, 36,37,38,39,40,41
Information gain, ranker	10	1,2, 40,3,30,7,33,40,21,20,34,11
CFS , best first,	5	1,2,7,8,33
Gain ratio and ranker	10	1,8,7,19,2,3,33,40,21,20,34,11
Chi Squared Eval + Ranker	10	1,2,7,3,40,33,19,34,30,29,21
Optimal Feature	9	1,2,7,33,3,40,34,30,21

From table 14, the optimum features to distinguish normal traffic with R2L attack include Duration, protocol type, Service, Land, Num_outbound_cmds, srv_serror_rate, Dst_host_count, dst_host_srv_count, dst_host_serror_rate

Table 15: features that distinguish between usual network traffic and User to Root (U2R) attacks.

Feature selection techniques	No of features	Selected Features
Original Dataset	41	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,31,33,34,35, 36,37,38,39,40,41
Information gain, ranker	10	11,40,3,30,10,29,14,1,33,21
CFS , best first,	4	6,11,29,30
Gain ratio and ranker	10	6,11,10,14,13,3,29,30,1,33
Chi Squared Eval + Ranker	10	6,11,3,10,14,40,30,29,31,1
Optimal Features	7	6,11,29,30,3,10,14

From table 15, the optimum features to distinguish normal traffic with U2R attack include Service, destination bytes, Hot, num_failed_logins, num_compromise, diff srv rate and srv_serror_rate

Table 16: The best set of relevant features

ATTACK TYPE	FEATURES	SELECTED FEATURES
ALL	8	2,3,4,26,27,36,39,41
DOS	9	2,3,9,26,41,4,26,27,41
PROBE	7	2,3,9,30,34,38,40
R2L	8	1,2,7,33,3,40,34,30,21
U2R	7	6,11,29,30,3,10,14
OPTIMAL FEATURE	12	1,2,3,9,26,27,29,30,34,36,39,40

The optimum features selected using the hybrid feature selection technique include: duration, src bytes, dst bytes, logged_in, same_srv_rate, diff_srv_rate, dst_host_count, dst_host_count, dst_host_srv, diff_host_rate, dst_host_srv_rerror_rate. Protocal_type, service, attck.

Table 17: Classification Results Using All Features of NSL KDD Dataset

Classifier	TP	FP	PR	RECALL	FM	ROC
J48	99	0.8	98	97	97	99
R Forest	99	0.8	96	98	98	99
Bayesian	97	1.7	97	97	97	99
PART	99	0.7	98	98	97	99

This work also compared the performance of proposed technique in terms of True Positive rate, false positive rate, precision, F measures and ROC with other schemes as indicated in Table 17. Comparing our proposed technique against using the full dataset with 42 features, the table indicates some enhancement has been obtained and even no degradation is observed. For example, the false positive and accurateness have been reduced by about 3% and enhanced around 5% respectively. Additionally, it is revealed that the anticipated system produced the optimum performance amongst other feature selection techniques.

Table 18: Classification Results Using Proposed Features of NSL KDD Dataset

Classifier	TP	FR	PR	RECALL	FM	ROC
J48	99	0.4	99	99	99	99
R Forest	99	0.2	99	99	99	1
Bayesian	95	1.6	96	95	96	99
PART	99	0.3	99	99	99	99

Table 18 and 19 determine the performance of various feature selection techniques assessed with diverse classification algorithms including: J48, Random forest, PART and Bayesian in relations to detection rate and false positive rate, respectively. From the tables, it can be revealed that, irrespective of the classification algorithms applied, the results from the anticipated method considerably increases comparing to other systems. For example, the detection rate from our planned algorithms is averagely 99% for all the classifiers as compared to the ALL, CFS, CHI and IG feature selection technique with average detection rate of 98 %, 98%, 91% AND 92% respectively.

Table 19: The Performance of Five Feature Selection Techniques with Different Classifier in Terms of Detection Rate

FST	J48	RF	PART	BAYES
Full	99	99	99	95
CFS	99	99	98	97
Chi square	92	93	93	88
IG	93	93	93	88
PROPOSED	99	99	99	97

Table 20: The performance of feature selection techniques with different classifier in terms of false positive rates

FST	J48	RF	PART	BAYESIAN
FULL	1	1	1	5
CFS	1	1	2	3
CHI	8	7	7	12
IG	7	7	7	12
PROPOSED	1	1	1	3

Table 20 indicates that the false alarm rate from the proposed algorithms has reduced significantly when compared with other systems. This assist to improve the performance. For example, on average the projected system with 1% has considerably reduced the false alarm rate than ALL, CFS, CHI and IG feature selection technique with 2, 2, 10 and 8 %, respectively.

Results Analysis and Discussion

As indicated by Table 14, the 41-features were reduced to 12-features. Features (6,11,29,30,3,10,14) are relevant for U2R, Features (1,2,7,33,3,40,34,30,21) are relevant for R2. Features (2,3,9,30,34,38,40) are relevant for PROBE class. Features (2,3,9,26,41,4,26,27,41) were selected as relevant DOS. The optimum features are 12 0ut of the total 42. They include duration, src bytes, dst bytes, logged_in, same_srv_rate, diff_srv_rate, dst_host_count, dst_host_count, dst_host_srv, diff_host_rate, dst_host_srv_rerror_rate. Protocal_type, service, attck.

The group with minimum applicable features to detect the attacks is the Content Based Features. These outputs are biased since the distribution of train and test datasets of U2R and R2L attacks is less recurrent as compared to the most frequent features from DoS and Probing attacks. This suggests that analyzing contents features from the network traffic packages is not compulsory to discover the attacks. The part of privacy and integrity from staffs is protected. The content feature may be from script in an email, and to accumulate and evaluate this type of information infringes the workers’ integrity.

The Basic Features are the best for examining to differentiate amongst usual network traffic and the attacks. These features define the amount of time for the connection, the protocol involved in the connection, the network service for the destination, normal or error status of the connection and the number of data bytes transferred from source and destination computer. The results show the importance of evaluating basic network traffic features to discover the attacks. The optimum features to discover DoS attacks includes Src_bytes, Diff_srv_rate, Service, Dst_bytes and Flag. The content features remain the least essential type of attributes to discover a DoS attack. This is because being that the DoS attacks generally comprise of either no information or occupied with a huge volume of impractical evidence.

The Host-based features like dst_host_srv_count, dst_host_serror_rate is important to detect probing attacks. These attacks take longer time and in different ports and also seek known vulnerabilities. For R2L attacks, the most important features duration includes, Src_bytes, Dst_bytes and srv_count. The duration represents the time in seconds for the connection and various R2L attacks have a period that is much longer as compared to an ordinary connection. The time-based feature, srv_count, which signify the amount of connections, have a low value as equated to normal network traffic. Throughout a R2L attack, the attacker attempts to gain entry to a local user account through a definite provision in connections extending more than 2 seconds.

To discover U2R attacks the most important attributes includes Service, num_failed_logins, root shell as the U2R attacks comprises the application of precise provision for remote access, mostly in combination using a file transfer service. Compared to the other types of attack, the content features are most significant to discover U2R attacks. The content features are generated by evaluating the content of a network connection. The significance of content features in discovering U2R attacks are as a result from the isolated operator’s activities that can only be observed while scrutinizing the content from the connection suites.

4.3 Enhanced Structural-Based Alert Correlation Method

The detection component of NIDSs generates a massive number of alerts and can overwhelm the network administrator. A computerized and intelligent grouping system is significant to discover their structural correlation through clustering alarms with similar features. The aim of this objective is to improve the Structural-based alert correlation model with machine learning technique to improve the worth of alarms and identify attack strategy. An innovative fusion based on clustering model is developed based on normalization, discretization and Improved Unit Range (IUR) technique to preprocess the dataset, EMFFS, Principal Component Analysis (PCA), SAC and proposed Post-Clustering algorithms is applied to condense the alarms complexity and improve the performance and unsupervised learning algorithm to aggregate similar alerts and to reduce the number of alerts. In the proposed model the performance of various unsupervised learning techniques like Self-organizing maps (SOM), Expectation Maximization, K-means, hybrid clustering and Fuzzy c-means (FCM) is compared.

In implementation of the model, the researcher used MATLAB Software. Three set of experiments were conducted and the results are tabulated in Table 21: In first experiment clustering with data preprocessing based on hybrid feature selection only (like., labeled as HFS), the second experiment clustering with PCA only (like., labeled as PCA), and the third experiment clustering with HFS and PCA (like., labeled as IPCA). The four measurements techniques applied are: (1) Clustering Error (CE) is the number of alerts that are wrongly clustered. (2) Error Rate (ER) is the percentage of wrongly clustered alerts, ER = (CE ÷ Total number of alerts observed) x 100, (3) Accuracy Rate (AR) is the percentage of alerts that are accurately clustered as they should be, AR = 100 – ER, and (4) Time is the algorithm processing time in seconds.

Algorithm Structural-Based Alert Correlation

```
Variables Di is a dataset for a network
(Ai) n x d is a n x d-dimensional alert instances for dataset Di n x d is n attributes with d alert instances Ci is an i-th cluster
DimRed is a variable for dimension reduction
ClusterAlert is a variable for clustering alerts
MF is a variable for merging and fusing redundant alerts
VP is a variable for verifying and prioritizing the alerts
F is a variable for filtering out the low quality alerts
Input Four set of dataset (Di), i = 1 to 4
Method
% Alert Clustering.
```

```
1. for each formatted and scaled dataset Di Select attributes to represent the alerts of (Ai) n x d which an alert is denoted Ai = {a1, a2, ..., an} where a is an attribute.
2. for each alert Ai in Di DimRed = PCA(Ai);
 3. ClusterAlert (Ai) = EM(DimRed);
% Alert Post-clustering. for each cluster Ci {
 4. % Alert Merging and Fusion Algorithm. MF = MergeFuse(ClusterAlert); % Subroutine MergeFuse.
 5. % Alert Verification and Prioritization Algorithm. VP = VerifyPrio(MF); % Subroutine VerifyPrio.
 6. % Alert Filtration Algorithm.  F = Filter(Ci);}
% Experimental Results with Standard Measurements.
 7. % Performance Validation. for all cluster Ci Calculate CE, ER, AR, Time;
 8. % Benchmark with other works. Repeat on other algorithms;
 9. % Save all results in database. return CE, ER, AR, Time;
Output Clustering Error (ER), Error Rate (ER), Accuracy Rate (AR), Execution Time (Time)
```

Table 21: Clustering Performance based on Self-organizing maps (SOM), Expectation Maximization, K-means and Fuzzy c-means (FCM)

Mode	FCM				K Means				SOM				EM			
	CE	ER	AR	TI	CE	ER	AR	TI	CE	ER	AR	TI	CE	ER	AR	TI
HFS	74	17.5	82.6	1.3	57	13.4	86.6	4.4	135	31.8	68.2	4.2	45	10.6	89.4	1.9
PCA	133	31.3	68.6	3.6	141	33.3	66.2	5.2	170	40.1	60.0	6.5	86	20.3	79.7	2.7
IPCA	67	15.8	84.2	4.8	46	10.9	89.2	6.2	112	26.4	73.6	7.4	41	9.7	90.3	4.6

The number of clusters in FCM, K-Means, and EM were varied to discover the best outcomes. The SOM was evaluated by concurrently changing the epochs and lattice configuration. Two third of the dataset were used for training and the rest for testing. The optimum result on SOM (73.58%) was obtained after being trained for 2500 epochs using hexagonal 4 by 6 lattice type and produced 12 clusters. The SOM's best processing time both for training and testing was obtained after 7.4 seconds. Increasing or decreasing the processing changes the results if the dataset, epochs and lattice type are greater (Siraj *et al.*, 2009c).The results of k-means clustering algorithm indicated that the performance depends on the number of clusters which are applied, and increasing or decreasing the cluster beyond the number of data types only lessens the efficiency of the model. Identifying the number of clusters therefore significantly changes the results.

The research has to determine the number of clusters that are expected in advance in order to obtain good results. In this work several clusters were tested and the optimum results (89.2) were obtained at 22 clusters in a time of 6.2 seconds. However, the challenge of identifying the number of clusters in a dynamic network, is much more difficult since there is no base data to assist in deciding the number of clusters.(Duque & Nizam, 2015). The best clustering algorithm was EM 90.3% and is arrived at 14 clusters in a time of 4.6 seconds. In respectively cluster, related alerts are clustered together and represent an attack step. The value of CE of FCM, K- Means, SOM and hybrid is larger, and hence a large number of alerts that belong together in one cluster are put into other different clusters. The result inferred that the proposed model based on hybrid feature selection, PCA and EM is effective in terms of clustering accuracy and processing time for this dataset.

4.4 Enhanced causal-based alert correlation model.

One of the major developments in machine learning in the past decade is the ensemble method, which generates highly accurate classifier by combining several classifiers generated from one or more learning algorithms. In this work a comprehensive analysis on prediction accuracy of three diverse ensemble techniques, bagging, boosting and stacking is performed in order to determine the algorithm with high detection accuracy and reduce false positive rate. Three different experiments on NSL KDD data set are conducted and their performance compared and evaluated based on accuracy, false alarms and computation time.

Experiment 1: Application of Bagging and Boosting with Five Classifiers

The experiment is conducted with two ensemble learning techniques, bagging and boosting and five classifier using 10-fold cross validation. The single classifier includes Bayes Net, IBK, Jrip and SVM and the results are illustrated in Table22. The conducted experiments are evaluated according to three performance measures which accuracy, false positive and execution time.

Table 22: The performance of bagging and boosting with five classifier using 10-fold cross validation

Algorithm	Accuracy			False Positive			Execution time	
	Single	Bagging	Boosting	Single	Bagging	Boosting	Bag	Boost
BayesNet	95.7%	95.5%	99.3%	4.3%	4.5%	0.670%	6.8	6.5
IBK	99.2%	99.1%	99.3%	0.80%	0.90%	0.7%	0.25	6.5
Jrip	99.5%	99.5%	99.5%	0.5%	0.5%	0.5%	395	390
J48	99.5%	99.5%	99.6%	0.5%	0.5%	0.4%	29.7	6.47
SVM	95.4%	90.53%	90.6%	4.6%	9.4%	9.47%	1656.8	1643.8

Comprehensively, all the systems attained acceptable outcomes, with the maximum accurateness of 99.6% and the minimum accurateness of 89.59%. Tables 21 indicates that Ad a boost if combined with J48 as a weak classification algorithm attains the maximum precision, of 99.6%, with a false positive (FP) level of 0.30%. however, the combination of Bayes Net and Bagging algorithm realizes the maximum FP rate of 4.5%. however, the calculation period from the three ensemble classification algorithms are completely very high; the slowest of all is the stacking algorithm and boosting and bagging follows respectively.

Table 22 indicates that the application of the bagging and boosting procedures have not increased the accurateness considerably. The application of boosting and the Bayes Net as a weak classifier algorithm improved the accurateness, by 3.6% only, whereas the rest produced not more than 1% improvement. Although the two ensemble systems were unsuccessful in improving the correctness, they were able to reduce the false positive rates. Bagging was successful in reducing the false positive rate with approximate 0.1% and 0.02% when implemented with IBK and Bayes Net respectively, boosting was able to reduce the false positive rate by up to 3.7% and 0.02% when implemented for Bayes Net and J48.

Many researchers compared the performance of bagging and boosting techniques with some comprehensive experimentations (Journal et al., 2014; Khorshid et al., 2015; Science, 2015; Sesmero, Ledezma, & Sanchis, 2015; Shrivastava, Baghel, & Gupta, 2013). The overall agreement of that boosting extends the minimum experimenting error have been considered the best precise existing mass-produced classification algorithm on a wide diversity of datasets. Nonetheless, it is detected that boosting algorithm are delicate to noise and outliers, specifically used for small datasets (Wahba et al., 2015). Bagging is effective with noisy data while boosting is relatively delicate to noise. Additional advantage of bagging systems is that both the training and classification phases can be conducted in parallel, while the boosting algorithm are done sequentially

Experiment 2: Application of stacking as a multi classifier with eight classifiers

In the stacking approach, the researcher compares eight diverse algorithms and SVM is the base classifier while stacking is a multi-classifier algorithm used. The researcher used several blends of Bayes Net, IBK, ANN, J48 and JRip. The classifications were predicted by the base learner's algorithm and was implemented as input variables into a stacking model learner. For each input classifier calculates expected classifications based on cross validation after which complete characteristic performance can be calculated. Afterwards the stacking model classifier trains with the data how to pool together the results from the dissimilar models to attain the best classification accurateness. The stacking algorithm testing outcomes are represented in Table 23.

Support Vector Machine, (SVM) is a classification algorithm applied for the classification, regression and outlier detection. The classification algorithm, is widely implemented in developing IDS due to its accurateness and robustness of outputs and its processing time to discover attacks (Chand et al., 2017; Khorshid *et al.,* 2015). The major features of SVM according to (Thaseen & Kumar, 2016) include: Deals with very large data sets efficiently, Multiclass classification can be done with any number of class labels, High dimensional data in both sparse and dense formats are supported, Expensive computing not required and can be applied in many applications like e-commerce, text classification, bioinformatics, banking and other areas. Even though SVMs are limited to making binary classifications, their superior properties of fast training, scalability and generalization capability give them an advantage in the intrusion detection application. Finding cost-efficient ways to speed up or parallelize the multiple runs of SVMs to make multi-class identification is also under investigation.

Table 23: The performance of SVM as a base learner and stacking as a multi classifier learner with seven classifier using 10-fold cross validation

Stacking met classifier	TP	FP	Precision	F measure	Execution time (sec)
SVM	96.4	0.029	96.1	96.1	762.33
SVM With Bayesian	98.9	0.7	98.9	98.9	21.8
SVM With RF	99.8	0.1	99.8	99.8	340.63
SVM With J48	99.8	0.1	99.8	99.8	40.1
SVM With ANN	93.6	6.4	93.5	93.7	1057.8
SVM With IBK	95.7	4.3	95.7	95.7	2147.1
SVM With Jrip	97.1	2.79	97.1	97.1	985
SVM With oneR	91.77	8.23	91.77	91.77	876

Bayesian is an extremely accessible classification algorithm and executes well while classifying coarse dataset example is medical data set. NSL-KDD data set is an already preprocessed data set, Bayes Net is consistent in achieving nearly the similar results in terms of accurateness, precision and recall of 98.9%. Whereas its false positive rate like. in correctively classified instances is 0.7%. The time to build the model is moderate at 21.8 seconds.

The JRip and OneR can be categorized as association rule mining algorithms. The JRiP processing speed is high and accurate algorithm while OneR generates a single rule in every feature and then selects the rule having minimum error (Hussain & Lalmuanawma, 2016; Song, 2016). Therefore, fusing SVM with JRip attains high accurateness and low positive values of. 97.21% and 2.79 % respectively whereas the OneR, it present high correctness and low false alarms figures of 91.77% and 8.23% respectively.

ANN is a robust classification algorithm and it also weak in learning. ANN needs huge data set for training (Amini, 2014). Hence as a result, this stacking with SVM does not provide very good performance. It produces better performance than SVM in terms of accurateness, false alarms, precision and recall.

The K - Nearest Neighbour (IBK) is a lazy learner. Hence it holds the training instances and perform actual work during the classification time (Jain & Rana, 2016). The IBK gives strongly consistent results. However, equal weightage is given to each of the attributes. consequently, fusing this classification algorithm with SVM produces moderately accurateness rate of. 95.79%. but slow in execution speed of 2147.1 sec.

The decision tree algorithms J48 and Random Forest provides high accuracy rate of 99.8 and a false positive rate of 0.1% and execution time of 40.1 seconds and 340.63 seconds respectively. Random Forest have the highest performance in terms of precision, recall, false positives and negatives validating metrics. NSL-KDD data set does not contain redundant records and it is easy for these classifiers to build their decision tree output and as a result combining them with SVM improves the overall performance of intrusion detection system.

Experiment 3: Hybrid machine learning model based on combination of five distinct classifiers based on voting and staking

The classification module of hybrid technique is composed of combination of five distinct classifiers based on stacking. This experiment is designed on the idea that each IDS is efficient in detecting a specific attack type. The techniques include Bayes Net, SVM, IBK, J48 and random forest with staking or voting as a multi classifier.

Table 24: Intrusion detection performance using combination of five distinct classifiers based on voting

Voting Rule	TP	FP	PRE	RECAL	F MEASURE	ROC	TIME
Product	99.5%	0.3%	99.5%	99.5%	99.9%	99.9%	1340.7
Majority	98.9%	0.9%	98.8%	98.9%	98.8%	99%	848.53

The results from Table 24 presents the results of classifier combination based on voting meta classifier. It provides more reliable results, as the final decision depends on the agreement amongst distinct classifiers. Improved detection accuracy and reduced error rates have been obtained by the fusion rule based on the product function paradigm because it takes into account the different discriminative power provided by the considered feature sets.

Table 25: Intrusion detection performance using combination of five distinct classifiers based on stacking with SVM as base classifier

Stacking	TP	FP	PREC	RECAL	FMEASURE	ROC	TIME
	99.8	0.1	99.8	99.8	99.8	99.9	4757.87

The results from Table 25 presents the results of classifier combination based on stacking meta classifier and SVM as base classifier. It provides more reliable results, as the final decision depends on the agreement amongst distinct classifiers.

It is observed that the voting and staking multi classifier techniques performs efficiently in terms of high detection rate, low false positive rate, precision and F measure. The time taken to build the model, is 4757.87 and 848.53 using staking and voting techniques respectively. In this thesis, voting technique is selected because of its high performance in detection rate and lower execution time.

4.5 Proposed Hybrid-Based Alert Correlation Model

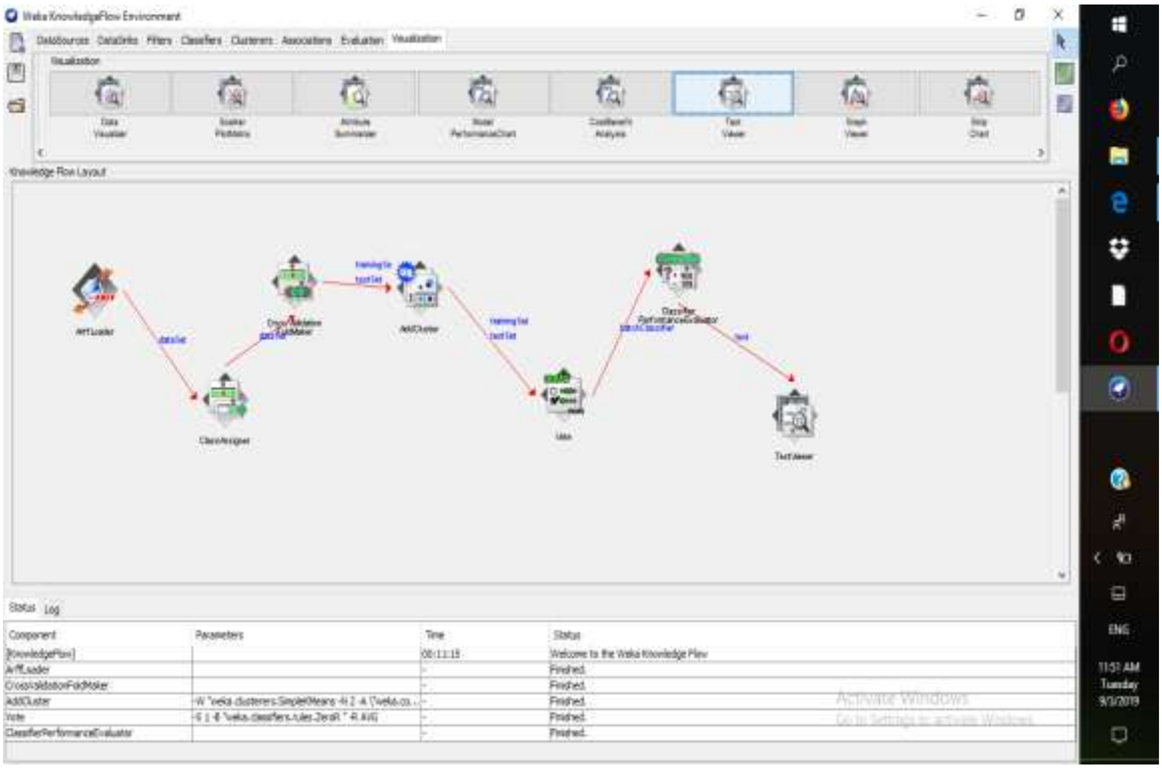
The proposed system hybridizes the artificial intelligent-based machine learning techniques to optimize the performance of the overall correlation and estimate the alerts attribute dependency. A new Hybrid-based AC (HAC) model that hybridizies the EMFFS, SAC and CAC models is designed. To accomplish such attempt, feature selection, PCA, EM algorithm, Post-Clustering algorithms, and voting algorithm are hybridized sequentially. The proposed system consists of four major phases as discussed in the paragraphs below.

In the first phase, a hybrid feature selection technique based on existing feature selections employed to find the best set of features to be used in this work. The Existing feature selection techniques includes Correlation Feature Selection (CFS) based evaluator with Best-first searching method, Gain Ratio (GR) Attributes based Evaluator with Ranker searching method, Information Gain (IG) based Attributes Evaluator with ranker searching method, and Chi Squared and Ranker searching method. When all the features of the input traffic are taken for processing, there is a lot of execution time and inaccurate output is produced. Experimenting with all the combinations of the features is exponentially complex in nature. Hence, only the relevant features are chosen and are given as input. The feature selection phase assists in drawing out the relevant features and as a result increases classifier accuracy, memory requirements and reduces computation speed. The optimum features are 12 out of the total 42. They include duration, src bytes, dst bytes, logged_in, same_srv_rate, diff_srv_rate, dst_host_count, dst_host_count, dst_host_srv, diff_host_rate, dst_host_srv_rerror_rate. Protocal_type, service, attck.

Dimension reduction reduces the alerts high-dimensionality for better classification performance while unsupervised learning algorithm clusters alerts into groups/attack steps to discover the structural correlation among the alerts. In this phase, the hybrid clustering model (PCA and EM) is implemented to discover the lists of attack steps and to improve the quality of alerts via filtering out a number of low-quality alerts (redundant alerts, false positives and low-risk alerts). The clustered alerts contain redundant and false positive alerts and hence, the alerts quality can be improved by removing such alerts via post-clustering. In order to identify the false positive alerts, the alert verification and prioritization algorithm is proposed to verify the alerts that are false positives and rank the rest of the alerts to determine their status and priority (whether High-risk, Medium-risk, Low-risk or Invalid). The alerts status and priority is essential to differentiate between more important (High-risk and Medium-risk) and less important alerts (Low-risk and Invalid). Such information would help Security Analyst in the decision making of reduction of alerts.

In the third phase, the output from the second phase which is the results from the hybrid clustering model (PCA and EM) is fed as input to the Multiple IDS Unit (MIU), and the output is the local decision (yi) derived from running different learning algorithms on the same data set. This section has five IDSs, each utilizing a unique algorithm is used independently for detecting a certain class of attack with improved accuracy, while performing moderately on the other classes. The five different types of IDS algorithms used are Support Vector Machines (SVM), IBK, Random Forest, J48, and Bayes Net and different results obtained and five outputs (local decisions) $y1$, $y2$ to $y5$ are obtained.

In the fourth phase, the output from each IDSi in MIU, considered as local decision (yi), is passed onto the multi classifier component based on majority voting rule and makes the final decision. Each classifier has a weight to denote the contributions of the classifier to the voting system. For each class to be identified, a weighted sum of base learners can be calculated. The output from each classifier is taken to the decision unit, and the global decision is taken based on the majority voting rule. If majority outputs from the MIU unit suggest Attack, then the decision unit decides that the input traffic is of ATTACK type; else it is NOT ATTACK. The approach based on classifier combination achieve effective attack detection as the combination of multiple evidences usually exhibits higher accuracies, like. lower false positives, than individual decisions. In addition, the generalization capabilities of pattern recognition algorithms allow for the detection of novel attacks that is not provided by rule-based signatures.



Component	Parameters	Time	Status
[KnowledgeFlow]		00:11:15	Welcome to the Weka Knowledge Flow
ArffLoader		-	Finished.
CrossValidationFoldMaker		-	Finished.
AddCluster	-W "weka.clusterers.SimpleKMeans -N 2 -A \"weka.co...	-	Finished.
Vote	-S 1 -B "weka.classifiers.rules.ZeroR " -R AVG	-	Finished.
ClassifierPerformanceEvaluator		-	Finished.

Figure 11: Proposed Hybrid-Based Alert Correlation Model

Algorithm: MIU

Variables Di is a dataset for a network

(Ai) n x d is a n x d-dimensional alert instances for dataset Di n x d is n attributes with d alert instances

Dim Red is a variable for dimension reduction

Ci is an i-th cluster

Cluster Alert is a variable for clustering alerts

MF is a variable for merging and fusing redundant alerts

VP is a variable for verifying and prioritizing the alerts

F is a variable for filtering out the low quality alerts

CLi is an i-th class

Classify Alert is a variable for clustering alerts

Input: Input traffic data record F {} set of all features

Output: Return whether traffic data record is (ATTACK or NOT an ATTACK)

Process:

a) for each IDMEF formatted and IUR scaled dataset Di,select attributes to represent the alerts of (Ai) n x d which an alert is denoted Ai = {a1, a2, …, an} where a is an attribute.

b) Identify the set of feature (F {}) to be used based on hybrid feature selection techniques (chi square, information gain and correlation based feature selection techniques).

c) for each alert Ai in Di Dim Red = PCA(Ai);

d) Pass the input traffic data record with f {} into clustering algorithm (EM), to aggregate similar alerts and filter out low quality alerts.

e) Post-Clustering algorithms. for each cluster Ci MF = Merge Fuse (Cluster Alert); % Subroutine Merge Fuse. VP = VerifyPrio(MF); % Subroutine VerifyPrio. F = Filter (Ci); % Subroutine Filter

f) Pass the results of clustering phase into classification algorithms SVM, IBK, J48, Random Forest and Bayes Net which returns the attack category for each input traffic data record.

g) For each input traffic data record, now there are five local decision $y1$, $y2$..., 5 from five classification algorithms.

h) The local decision yi is labeled as $yy1$ or $yy2$ $yy1$—stands for ATTACK $yy2$—stands for NOT an ATTACK

i) If (yi == “DOS” ‖yi == “PROBE” ‖yi == “U2R” ‖yi == “R2L”) Then yi = $yy1$ Else yi = $yy2$

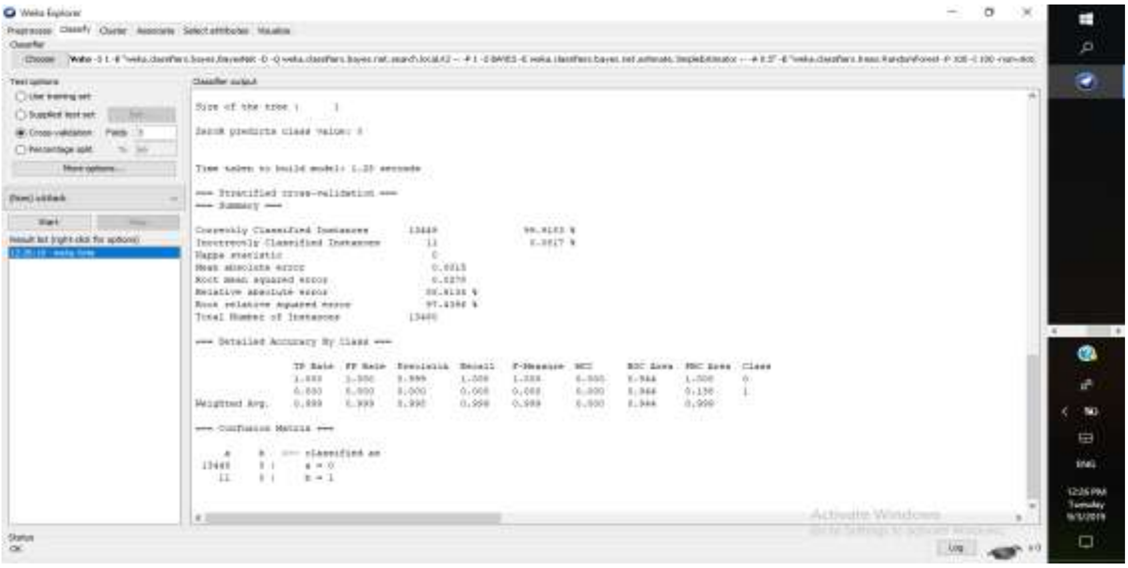
j) For each input traffic data record, decision from five IDS units is either $yy1$ or $yy2$ count the number of $yy1$ and $yy2$ If ($yy1$ > 3) Final decision = $yy1$ Else Final decision = $yy2$

4.6 Performance Evaluation and Results

The experimental results and its discussions are presented as follows. Section 4.5.1 shows results based on performance metrics and Section 4.5.2 represents comparison results of proposed model and other models.

1.8.23 4.6.1 Evaluation based on Performance Metrics

The performance of the proposed intrusion detection system is evaluated with the help of confusion matrix. The measurements metrics are true positives (TP), false positives (FP), Precision (P), Recall (R), F-measure (FM), Receiver Operating Characteristic (ROC) Area and Execution Time (Time). The results are shown in figure 12 below.



Vote -S 1 -B "weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5" -B "weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots

```
Size of the tree :      1

ZeroR predicts class value: 0


Time taken to build model: 1.25 seconds

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

Correctly Classified Instances       13449               99.9183 %
Incorrectly Classified Instances        11                0.0817 %
Kappa statistic                          0
Mean absolute error                      0.0015
Root mean squared error                  0.0278
Relative absolute error                 88.8138 %
Root relative squared error             97.4386 %
Total Number of Instances            13460

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.999      1.000    1.000      0.000    0.944     1.000     0
                 0.000    0.000    0.000      0.000    0.000      0.000    0.944     0.136     1
Weighted Avg.    0.999    0.999    0.998      0.999    0.999      0.000    0.944     0.999

=== Confusion Matrix ===

     a     b   <-- classified as
 13449     0 |     a = 0
    11     0 |     b = 1
```

Figure 12: Performance of Proposed Hybrid-Based Alert Correlation Model

The value of TP, P, R, FM and ROC are 0.998, 0.99, 0.99 and 0.94 that close to ‘1’ indicates excellent performance and below ‘0.5’ indicates average or bad performance. A smaller value of FP (close to zero) shows good performance since the amount of false classification is very small. The time taken to build the model is only 1340.7seconds.

1.8.24 4.6.2 Comparison Results Between Proposed HAC Models and other Models.

The experimental results of Thomas and Balakrishnan (2007), Kaliappan, Thiagarajan, & Sundararajan, (2015), paper are taken for a comparative study. Kaliappan *et al.*,(2015) designed a model Fusion of Heterogeneous Intrusion Detection Systems for Network Attack based on the idea that each IDS is efficient in detecting a specific type of attack. The feature selection is done with the help of genetic algorithm the proposed and Multiple IDS Unit (MIU), consist of five IDS units Support Vector Machines (SVM), IBK, Random Forest, J48, and Bayes Net., and each IDS follows a unique algorithm to detect attacks. The selected features of the input traffic are passed on to the MIU for processing. The decision from each IDS is termed as local decision. The fusion unit inside the MIU processes all the local decisions with the help of majority voting rule and makes the final decision.

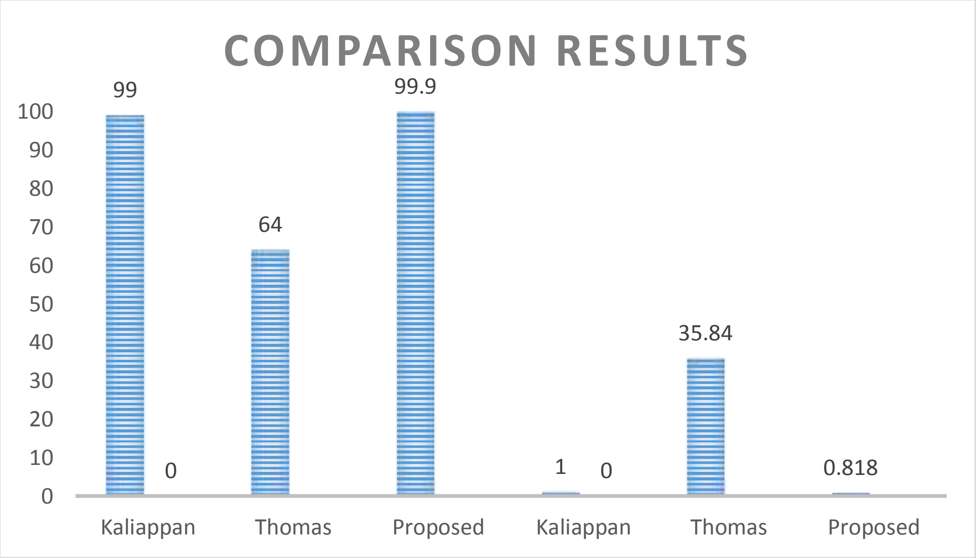
Thomas and Balakrishnan (2009) have optimized the performance of IDS using fusion of multiple IDS. The assignment of weight for each IDS is outlined in this paper, and the weights are aggregated to take a correct decision. DARPA 1999 data set is used to evaluate the IDSs which are outdated. It contains more redundant records, and so it affects classifier accuracy. In their method, binary values are used to decide attack or normal.

Table 26: Comparison results of detection rate and false alarm rate for Thomas and Balakrishnan [2009] work, Kaliappan et al., (2015) and proposed model for different attack

	Detection rate			False alarm rate		
Attack	Kaliappan	Thomas	Proposed	Kaliappan	Thomas	Proposed
DOS	99	64	99.8	1	36.5	0.094
PROBE	99	76	99.8	1	24,32	0.011
U2R	98	92	99.7	1/38	8.1	0.818
ALL	99	64	99.9	1	35.84	0.818

From Table 26 the detection rate for DOS is 64% and 99% for the Thomas and Balakrishnan (2009) work, Kaliappan et al.,(2015) and 99.8 for the proposed system. Likewise, there is an improvement in detection rate in PROBE, U2R, and R2L, as compared with work for the

Thomas and Balakrishnan (2009) work, Kaliappan et al.,(2015) with the highest improvement in the detection rate of R2L. The false alarm rate for DOS is 36.20 and 1.0 in the work of Thomas and Balakrishnan (2009) and Kaliappan et al.,(2015) respectively, but in the proposed work, the value is minimized to 0.094 and for PROBE, U2R, and R2L also the false alarm rate has decreased drastically.



Detection rate false alarm rate

Figure 13: Comparison results of detection rate and false alarm rate for Thomas and Balakrishnan (2009) work, Kaliappan et al.,(2015) and proposed model

Overall the results show the effectiveness of classifier combination in providing more reliable results like. the detection rate is ,99.9% while the false detection rate is 0.1%. The proposed system has better results accurate rate of 99.9% and false positive rate of 0.1% compared to detection rate of 99% and 64% and 1% and 35.84% for kaliapan (2015) and Thomas (2009) respectively.

4.7 Evaluation with UNSW-NB15 datasets

A similar experiment was conducted using UNSW-NB15 datasets and the results compared with the findings of previous work using NSL KDD dataset. The UNSW-NB15 (Janarthanan, 2017) dataset was published in 2015 by (Moustafa & Slay, 2014, 2015b) for research purposes in IDS. It is a hybrid of attack activities include real traffic and synthesized activities in a computer network traffics and comprises of nine different moderns attack types as compared to fourteen (14) attack types in KDD’99 datasets activities of normal traffic that

were captured with the change over time. The UNSW-NB15 dataset has forty-nine (49) features that comprised the flow based between hosts (like., client-to-server or server-to-client) and the packet header which covers in-depth characteristics of the network traffic.
The UNSW-NB15 dataset has been divided into two Training datasets (82, 332 records) and a Testing dataset (175, 341 records) including all attack types and normal traffic records. Both the Training and Testing datasets have 45 features. The features scrip, sport, dstip, stime and ltime are missing in the Training and Testing dataset.

The feature selection techniques help to identify some of the important attributes in a data set, in order to obtain the best features to make a classifier algorithm. In order to find the best subset of features in WEKA, a few methods of Attribute Selection were employed against UNSWNB15 dataset such as Cfs Subset Eval (attribute evaluator) + Greedy Stepwise method and InfoGain Attibute Eval (attribute evaluator) + Ranker method. Then we train and test with the supervised and unsupervised machine learning algorithms. Clustering the training dataset is done based on a new hybrid clustering model based on PCA and EM clustering algorithm and classification The classification module of hybrid technique is composed of combination of four distinct classifiers based on stacking Ensemble learning technique. The five techniques used are Bayes Net (NB), Decision Tree (DT) like J48 and random forest, Support Vector Machine (SVM), and Instance Based Learners (IBK). The techniques used to evaluate the performance of the proposed model are in terms of accuracy and false alarm rates (FAR), Stratified Cross Validation of 10-fold on the UNSW-NB15 data set.

1.8.25 4.7.1 Enhancing Feature Selection based on Hybrid Feature selection technique

This section analysed the features included in the UNSW-NB15 dataset by employing machine learning techniques and exploring significant features (curse of high dimensionality) by which intrusion detection can be improved in network systems. The feature selection techniques to be employed include correlation, information Gain and Chi square. Data preprocessig was impleted based on normalisation, improved unit range and principle component analysis to reduce the dimensionality of the data set.

Table 27: UNSW-NB15 Features of the Proposed Model

CATEGORY	FEATURE NUMBERS
NORMAL	11,34,19,20,21,37,6,10,11,36,47
DOS	6,11,15 16,36,37,39,40,42,44,45
FUZZERS	6,11,14,15,16,36,37,39,40,41,42
BACKDOORS	6,10,11,14,15,16,37,41,42,44,45
EXPLOITS	10,41,42,6,37,46,11,19,36,5,45
ANALYSIS	6,10,11,12,13,14,15,16,34,35,37
GENERIC	6,9,10,11,12,13,15,16,17,18,20
RECONNAISSANCE	10,14,37,41,42,43,44,9,16,17,28
SHELLCODE	6,9,10,12,13,14,15,16,17,18,23
WORMS	41,37,9,11,10,46,23,17,14,5,13
COMMON	6,9,10,11,12,13,14,15,16.17,36,37,41,42,44,45

The researcher computed the final results depending on the highest repeated features with at least three times. The hybrid features to be applied to develop the model are defined in (Moustafa & Slay, 2015) as Service type (e.g. http, ftp, smtp, …etc) [service], Source to destination bytes [sbytes], Source to destination time to live [sttl], Mean of packet size transmitted by the srcip [smean] and No. of rows of the same dstip and the sport in 100 rows [ct_dst_sport_ltm]. Features service (Janarthanan, 2017), sbytes, and sttl are from the Basic Feature category. Feature smean is from the Content Features category and feature ct_dst_sport_ltm is from Additional Generated Features category.

1.8.26 4.7.2 Intrusion detection performance based on individual classifiers on UNSW-NB15 dataset

The evaluation criteria of the proposed hybrid feature selection techniques applying the Bayesian Net, IBK. PART, J48, Random Forest and SVM are computed in terms of accuracy (Acc.) and False Alarm Rate (FAR) to evaluate the complexity of these data sets.

Table 28: Classification Results five different classifiers on UNSW-NB15 Dataset

Technique	TP	FP	PR	RECALL	F MEASURE	ROC	TIME (sec)
R. Forest	0.987	0.023	0.987	0.987	0.987	0.99	16.36
Bayes Net	0.926	0.135	0.927	0.926	0.924	0.97	2.66
Part	0.946	0.099	0.947	0.946	0.945	0.99	40.67
J48	0.953	0.07	0.953	0.953	0.953	0.986	13.73
IBK	0.946	0.073	0.946	0.946	0.946	0.966	0.04
SVM	0.8467	0.18	0.85	0.84	0.847	0.833	953.89

The evaluation criteria of the UNSW-NB15 data set show that, the existing algorithms of the decision engine True positive rates are 98.7%, 92.6%, 94.6%, 95.3%, 73% and 84.67 % for Random forest, Bayes net, Part. J48, IBK and SVM respectively. The false positive rate is 02.3%, 13.5%, 9.9% and 0.07%, 7.3% for Random forest, Bayes net, Part, J48, IBK and 18% respectively. This indicates that individual algorithms cannot detect new patterns which are able to discriminate between the similar record values of each feature.

1.8.27 4.7.3 Intrusion detection performance using combination of five distinct classifiers

The classification module of hybrid technique is composed of combination of five distinct classifiers based on stacking. The theory behind designing the experiment is that each IDS is effective in identifying a particular attack type. The techniques include Bayes Net, SVM, IBK, J48 and random forest with staking or voting as a multi classifier.

Table 29: Intrusion detection performance using combination of four distinct classifiers (Random forest, Bayes net, J48, IBK) based on stacking with SVM as base classifier

Stacking	TP	FP	PREC	RECAL	FMEASURE	ROC	TIME
	0.955	0.068	0.955	0.955	0.955	0.944	3732.9

The classification module of the hybrid technique is a combination of four distinct classifiers based on stacking. The techniques include Bayes Net, IBK, J48 and random forest with staking as a multi classifier and SVM as base classifier. On average, 95.5% of detection rate is achieved and the false alarm rate of 0.068 is achieved. The execution time for the proposed system is 3732.9 seconds.

1.8.28 4.7.4 Comparison of proposed model based on NSL KDD dataset and UNSW-NB15 dataset

The proposed hybrid model is able to improve the detection accuracy, decrease the false alarm rate by reducing the processing time. The detection rate, false alarm and execution time using NSL KDD data set is 99.9, 0.818, 0.999 and 1340.7respectively while using UNSW-NB15 is 95.5, 6.8 and 3732.9 respectively. The results show that, the evaluation of the NSLKDD data set is better than the. UNSW-NB15 in detection accuracy and the false alarm rate but the processing time is higher. 0.999

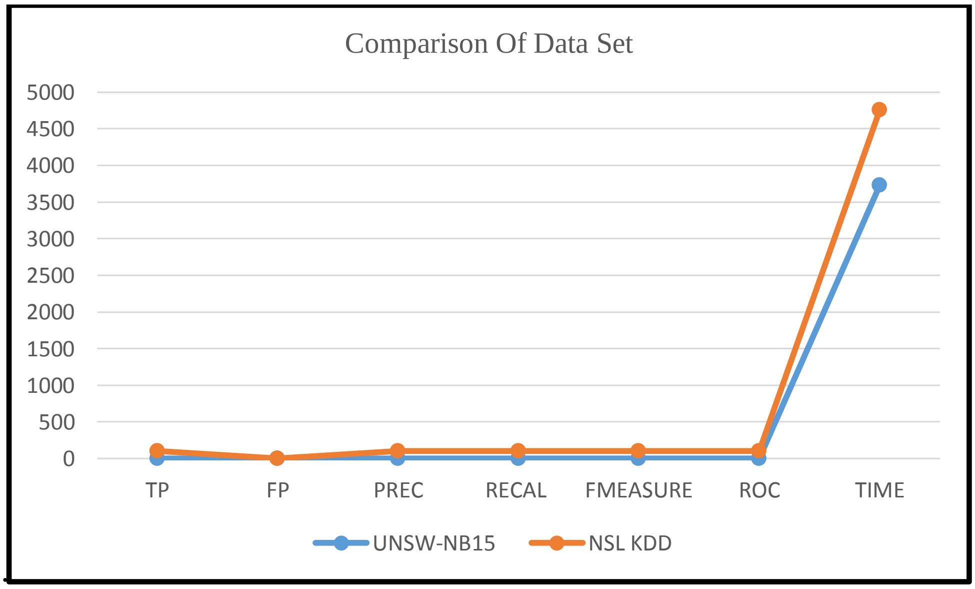


Figure 14: Evaluation of the NSLKDD data set and UNSW-NB15 in detection accuracy and the false alarm

4.8 Summary

In this chapter, a new Hybrid-based AC (HAC) model that hybridizes preprocessing techniques, ensemble feature selection technique, dimensional reduction technique, clustering algorithm, post-clustering algorithms, classification algorithm and statistical correlation tests has been proposed and discussed. It discovers correlation in terms of feature selection, structural, causal as well as statistical correlation. In other words, the EMFFS, SAC, CAC and StAC are hybridized in a new HAC model. The purpose of such sequential hybridization is to improve the alert classification performance as well as the overall correlation performance. The experimental results show that our HAC model produces better classification and overall correlation performances than other current than other current works. This improvement can be achieved since the voting classification is performed on balanced, dimensionally reduced and improved alerts via implementation of hybrid feature selection, PCA, EM and post-clustering algorithms.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

This chapter summarizes and concludes the findings of the study. It also lists out the general and specific contributions in terms of philosophy, knowledge and design. Some potential works are also suggested and recommended for further research and extension.

5.2 Summary

This study was set out to improve the performance of Network Intrusion Detection System (NIDS) by developing an alert correlation model based on hybridized machine learning techniques. Innovative and novel alert correlation models are designed to accomplish the philosophy of "providing a complete and optimal alert correlation".

1.8.29 5.2.1 To identify the optimum features based on hybrid feature selection techniques

The main aim of feature selection is to eliminate irrelevant and repetitive features from the dataset to make robust, efficient, accurate and lightweight intrusion detection system. To achieve this objective, an Enhanced Feature Selection Based On Multi-Filter Feature Selection (**EMFFS**) Method is developed to find the best set of features that are used in this work. The feature selection techniques integrated are Correlation Feature Selection (CFS) based evaluator with Best-first searching method, Information Gain (IG) based Attributes Evaluator with ranker searching method, and Chi Squared and Ranker searching method.

The NSL KDD data set features are reduced from total 42to 12 optimum features. They include duration, src bytes, dst bytes, logged_in, same_srv_rate, diff_srv_rate, dst_host_count, dst_host_count, dst_host_srv, diff_host_rate, dst_host_srv_rerror_rate. Protocal_type, service, attck. The EMFFS method is evaluated with several feature selection algorithms including: CFS, IG, and chi-squared on NSL-KDD dataset. The outcomes signify that the proposed system has significantly reduced training time and increased the accurateness and precision. Also, to demonstrate the effect of pre-processing dataset on classification rate using filter feature selection methods, the feature selection methods are evaluated with four diverse classification algorithms including: J48, Random forest, PART and Bayesian based on detection rate and False Alarm Rate. Nevertheless, the outcomes suggest that the proposed system produces better results as compared with other techniques. Additional remark based on the evaluation outcomes amongst the proposed system and employing the full dataset is that J48 classification algorithm have better results with proposed feature selection algorithm as compared to other classification algorithms. This is expected as random forest is an ensemble classifier that combines a collection of classifiers to make a forest. Results showed that the proposed feature selection model could assist in building lightweight IDS that maintains high detection rates with a fast and reliable training and testing while consuming less system resource. The effectiveness and feasibility was verified by several experiments the results indicate that, the enhanced model is not only able to yield high detection rates but also able to speed up the detection process.

1.8.30 5.2.2 To enhance the structural based alert correlation model using unsupervised machine learning techniques.

The IDS produce voluminous alerts which contains low level evidence which are time consuming. tedious and lab our intensive when analyzed manually and hence alert clustering and analysis is crucial as alerts are insignificant when they are independent, therefore discovering the associations amongst them is an essential phase. Aggregating and grouping the alarms based on the similarities of attributes will discover the attack strategy used by the attackers. Additionally, duplicated alerts can be identified and combined. This work compared four unsupervised algorithms such as Self-organizing maps (SOM), K-means, and Fuzzy c-means (FCM) and Expectation and maximization (EM) technique to find which algorithm have high clustering accuracy rate and low processing time. The output is a fusion of artificial intelligent method for computerized alert clustering and filtering in intrusion alert analysis based on EMFFS, Principal Component Analysis (PCA) and Expectation and maximization techniques that gives optimum results to aggregate similar alerts and to reduce the number of alerts. The results are promising in terms of clustering accuracy rate (89.2) and processing time (6.2 sec) but it cannot reveal the memberships of attack stages which is also important in determining the attack strategy.

1.8.31 5.2.3 To enhance the causal-based alert correlation model using supervised machine learning techniques.

Artificial based systems and their ensembles are currently inviting significant consideration from the research community for intrusion detection. Their features, such as flexibility, adaptability, new pattern recognition, fault tolerance, learning capabilities, high computational speed, and error resilience for noisy data, fit the prerequisite of building effective IDS. Ensemble approach imitates our second nature to look for several opinions before making an essential decision. The basic principle is to evaluate several individual pattern classifiers, and integrate them in order to reach a classification that is better than the one obtained by each of them separately.

Support Vector Machine (SVM) is a machine learning technique with high leaning capability, detection level and execution speed and applicable specifically for intrusion detection. Nevertheless, its performance can be considerably enhanced if combined with other classification algorithm. In the first experiment, the researcher conducted an investigation to compare the performance of SVM classification algorithm when stacked using other algorithms including Bayes Net, AdaBoost, bagging, Artificial Neural Networks (ANN), IBK, J48, Random Forest, Jrip and OneR. Ensemble algorithm produce improved classification results in comparison to an individual classification algorithm especially for discovering low occurrence attacks for instance U2R and R2L. The main objective is to establish the classification algorithm that produce the optimum output when detecting the intrusions and incase if combined with SVM. An anomaly detection module is developed by integrating Support Vector Machine (SVM) as a base classifier with other machine learning algorithms using stacking as a Multi-Classifier and tested their performance on NSL-KDD data set based on the precision, recall, false alarm rate and accuracy. The stacking of SVM and Random Forest outperforms other classifiers for the considered data-set and parameters with the accuracy of 99%. The second experiment examines the likelihood of employing ensemble algorithms to enhance the productivity of NIDS.

The research experimented with three diverse ensemble classifiers, bagging, boosting and stacking, with an objective to enhance the accurateness and decrease the false positive rate. several artificial intelligence individual algorithms including, Bayes net, J48 (decision tree), JRip (rule induction) and IBK (nearest neighbour), were applied as base classifiers to the ensemble techniques using 10-fold cross validation. Overall the application of bagging and boosting did not significantly improve the accuracy or reduce error rates and only in Stacking technique a reduction of false positive rate of 3% was achieved. The key assumption that the errors due to the individual models are uncorrelated is unrealistic; in practice, the errors are typically highly correlated, so the reduction in overall error is generally small (Govindarajan, 2016; Sesmero et al., 2015). Staking method is slow in execution as compared with the other ensemble learning technique. The random forest classifier outperformed the three other methods (Bayes net, J48 (decision tree), JRip (rule induction) and IBK (nearest neighbour),) and attained the maximum accurateness and the minimum false positive rate, and fast in processing speed.

1.8.32 5.2.4 To design an alert correlation model based on hybrid machine learning techniques to enhance the performance of network Intrusion Detection Systems.

In order to improve the ability to detect all the intrusions (known and unknown) and reduce false positive rate and execution time, this research propose to develop a hybrid anomaly detection module by combining heterogeneous classifiers based on majority voting rule to arrive at the optimal decision.

To design an alert correlation model based on hybrid machine learning techniques to enhance the performance of network Intrusion Detection Systems. The major correlation stages include feature selection, enhanced structural based correlation, enhanced causal based correlation modules and heterogeneous fusion of different classifiers.

The proposed model based on EMFFS uses only relevant features derived from NSL KDD data set as the input traffic data for processing (src bytes, dst bytes, logged_in, same_srv_rate, diff_srv_rate, dst_host_count, dst_host_count, dst_host_srv, diff_host_rate, dst_host_srv_rerror_rate. Protocal_type, service, attck.). The clustering phase employed the principal component analysis, Expected and maximization (EM) to group related alerts and to remove the complexity of alerts. The classification module is designed based on the fact that every classification algorithm is competent in discovering a definite attack type. The central component in this module is the Multiple IDS Unit (MIU) which combines different individual classification algorithm including SVM, Bayes Net, IBK, J48 and random forest as based classifier with staking as multi classifier. It provides more reliable results, as the final decision depends on the agreement amongst distinct classifiers.

Table 30: Intrusion detection performance using combination of five distinct classifiers based on voting

Voting Rule	TP	FP	PRE	RECAL	F MEASURE	ROC	TIME
Product	99.9%	0.3%	99.9%	99.9%	99.9%	99.9%	1340.7
Majority	98.9%	0.9%	98.8%	98.9%	98.8%	99%	848.53

It is observed that the voting multi classifier techniques performs efficiently in terms of high detection rate, low false positive rate, precision and F measure. The execution time for the model, is 1340.7and 848.53 using product rule and majority techniques respectively. In this thesis, voting technique is selected because of its high performance in detection rate and lower execution time.

In summary, a new Hybrid-based AC (HAC) model that hybridizes ensemble feature selection technique, dimensional reduction technique, enhanced structured based techniques, enhanced correlation-based technique and statistical correlation tests has been proposed and discussed. The proposed technique includes EMFFS method, Principal Component Analysis (PCA), Expectation and Maximization (EM) techniques and Voting multiclassifier technique with five classifiers.

1.8.33 5.2.5 To validate the model based on derived metrics and comparisons with current alert correlation models

The experimental results and its discussions are based on performance metrics and a comparison results between proposed model and other models. The measurements metrics are true positives, false positives and Execution Time and the experimental results are 99.9, 0.1 and 1340.7 seconds respectively.

An investigation to compare the detection rate and false alarm rate of the proposed and existing fusion methods was conducted. when compared with the work of Thomas *et al* (2009) and Kalipan *et al* (2015), an improvement in the detection rate and false alarm rate is achieved. The false alarm rate for DOS is 36.20 in the work of Thomas at el (2009), but in the proposed work, the value is minimized to 1.0 and for PROBE, U2R, and R2L also the false alarm rate value has decreased to 0.01, 0.818. and 0.818 respectively. Similar experiment was conducted to get the performance of proposed NIDS using UNSW-NB15 dataset in terms of feature characteristics to distinguish between normal and abnormal records. The hybrid features to be applied to develop the model are defined are identified based on hybrid feature selection technique comprising Correlation Feature Selection CFS, information gain (IF) and chi square. Similar features with Moustafa *et al*, (2015) was obtained. They include Service type (e.g. http, ftp, smtp, …etc.) [service], Source to destination bytes [sbytes], Source to destination time to live [sttl], Mean of packet size transmitted by the srcip [smean] and No. of rows of the same dstip and the sport in 100 rows [ct_dst_sport_ltm]. Features service (Janarthanan, 2017), sbytes, and sttl are from the Basic Feature category. Feature smean is from the Content Features category and feature ct_dst_sport_ltm is from Additional Generated Features category.

The decision engine of the NIDS employs Bayes Net, IBK. PART, J48, Random Forest with staking or voting as a multi classifier. Our research evaluates the complexity of the two data sets in terms of accuracy and false alarm rate. The results show that, the evaluation of the NSLKDD data set (99%) is better than the. UNSW-NB15 (95.5%) in detection accuracy and the false alarm rate but the processing time is lower (3732.9 sec) than (4757.87) of NSLKDD data set. However, the evaluation criteria of the UNSW-NB15 data set show that, the existing algorithms of the decision engine cannot detect several records categories, because of the similarities between the values of these records.

In summary, the proposed model is accurate and has low false positive rate though the execution time in its application in the intrusion detection field is higher. This work provides a good reference point for further research in alert correlation and computer forensic investigation.

5.3 Conclusion

This study has proposed a new AC model for analyzing alerts known as HAC which integrates the three correlation models (feature selection, SAC and CAC) using hybrid intelligent-based architecture. Since sufficient knowledge on AC research is much needed, this thesis also has described the related literature review on current works and techniques as well as detail methodology on designing the HAC. In order to discuss HAC in depth, new framework and related algorithms have been proposed and evaluated. The idea of hybridizing multiple types of correlations is due to several problems in the area of information security for Intrusion Detection System (IDS):

i. The implementation of multiple intrusions sensors or NIDSs in the same environment of networks leads to the problem of managing data. The large volume alert with low quality of alerts which includes the unformatted or multi-formatted alerts, false positive alerts, redundant alerts and low risk alerts. Such problem could affect badly the accurateness and effectiveness of the alert analysis.
ii. The evolving or creation of new multi-stages attack strategy from time to time causes the emerging of new attack steps and stages. This problem makes the rule-based and condition matching-based AC models are unpractical and unreliable.
iii. Existing AC models are having limitations of providing only single correlation, depending on rules or expert knowledge and requiring manual parameters setting.

Since all these problems need to be addressed and solved simultaneously, thus a hybrid solution is essential. Potential advantages of sensor alert fusion include elimination or reduction of the need of manual analysis of reported data, compression of alert volume and identification of context by associating alerts from different sensors. The aim is to offer a complete analysis of alerts that are generated by multiple sensors/NIDSs by considering a complete view or coverage of correlation. An effective and practical AC model that can provide knowledge on attack strategy is very important and useful to SA for designing and proposing response mechanisms precisely. Therefore, the scope of attack strategy offered by HAC includes improving the quality of alerts, identification of attack steps and recognition of known and new attack stages. For validation and benchmark purpose, HAC is experimented using NSL KDD Specific dataset. The experimental results have shown that HAC (through its proposed model) can effectively discover better correlation completeness compared to current works.

5.4 Contributions

The main contributions by this research can be broadly be classified into

i. A new insight to offer a complete coverage of alert correlation has been proposed. It is challenging to completely analyze and understand the alerts that are generated from multiple locations of NIDSs on the networks. From the human perspective, a complete diagnosis is urgently needed in order to propose a suitable and effective cure. It does the same in the information security perspective, especially in proposing the preventive and responsive actions toward network attacks based on the results presented by AC model.

ii. Enhanced and new correlation models that offer complete correlation in terms of structural, causal and statistical with optimal performance. This research also proposed three enhanced and new AC models based on feature selection, unsupervised learning, supervised learning and HAC based on integration of the three learning algorithms as well as statistical correlation tests to address the problem of low-quality alerts and recognizing known and new patterns of alerts.

5.5 Recommendations for further research

This dissertation introduces several directions for future research.

i. Different type of sensors. Alerts from different types of NIDSs are worth to be investigated. Since this research applied multiple signature-based NIDSs, the performance using anomaly-based NIDSs is unpredictable. On top of that, it would become an interesting yet challenging research if alerts are taken from both types of NIDSs (signature-based and anomaly based).

ii. Real-time and on-line alert analysis. Using benchmark dataset, this research is restricted to an off-line correlation for the alert analysis. furthermore, an on-line correlation can offer a more reliable analysis since the responsive strategy can be planned earlier. But, the main challenge is how to ensure that the same correlation model is not performed repeatedly in a very short period of time. Additionally, actual network attacks may fail the operating systems and true alerts can be missed unless they are being logged. Other challenge includes the determination on sliding window or ‘waiting’ period that is needed by AC model to wait and gather maximum number of alerts to be correlated.

iii. Combination of other supervised and unsupervised learning algorithms. There exists no single model that can fit all dataset and problems in all area of researches. If possible, this research can be extended with other kind of learning algorithms in providing a generic model so that it can fit to any other information security dataset.

iv. Investigation on applying Independent Component Analysis (ICA). ICA is another extraction algorithm that independently characterizes or chooses the significant components from the dataset. It has similar function as PCA which can reduce the alerts dimensionality to improve the correlation performances. In fact, investigation on combining PCA and ICA is also a brilliant idea.

v. The testing phase of intrusion detection correlation systems is not standardized as the developers of different systems employ different testing methodologies. As a result, it is difficult to compare the performance of two systems and most sensors are not publicly available and hence it is not feasible to perform a side by-side comparison test. The research in the development of a standardized test could help solve this problem.

5.6 Conclusion

The goal and objectives of this research have been successfully achieved. In summary, it has contributed new philosophy on providing complete alert correlation by proposing a new Hybrid-based AC (HAC) model. With HAC, SA gains the knowledge on list of attack steps, membership of attack stages and the attributes dependency strength among the alerts. In fact, this knowledge leads to the understanding of the attack strategy launched by the attacker that could assist SA in designing effective respond and preventive mechanism. It also has opened up new research opportunities and ideas in the area of AC research. Hopefully, the contributions offered are advantageous and inspirational.

REFERENCES

Akande, K. O., Owolabi, T. O., Twaha, S., & Olatunji, S. O. (2014). Performance Comparison of SVM and ANN in Predicting Compressive Strength of Concrete, *IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 5, Ver. I (Sep – Oct. 2014), PP 88-94 www.iosrjournals.org*

Albayati, M., & Issac, B. (2015). Analysis of Intelligent Classifiers and Enhancing the Detection Accuracy for Intrusion Detection System. *International Journal of Computational Intelligence Systems*, *8*(5), 841–853. Retrieved from :https://doi.org/10.1080/18756891.2015.1084705

Alhaj, T. A., Siraj, M. M., Zainal, A., Elshoush, H. T., & Elhaj, F. (2016). Feature selection using information gain for improved structural-based alert correlation. *PLoS ONE*, *11*(11), 1–18. Retrieved from https://doi.org/10.1371/journal.pone.0166017

AliShah, A., Sikander Hayat Khiyal, M., & Daud Awan, M. (2015). Analysis of Machine Learning Techniques for Intrusion Detection System: A Review. *International Journal of Computer Applications*, *119*(3), 19–29. Retrieved from https://doi.org/10.5120/21047-3678

Ambusaidi, M. A., He, X., & Nanda, P. (2015). Unsupervised Feature Selection Method for Intrusion Detection System. *Trustcom/BigDataSE/ISPA, 2015 IEEE*, *1*, 295–301. Retrieved from https://doi.org/10.1109/Trustcom.2015.387

Amini, M. (2014). Effective Intrusion Detection with a Neural Network Ensemble Using Fuzzy Clustering and Stacking Combination Method, Journal of Computing and Security*1*(4), 293–305.

Assi, J. H., & Sadiq, A. T. (2017). NSL-KDD dataset Classification Using Five Classification Methods and Three Feature Selection Strategies, Journal of Advanced Computer Science and Technology Research, Vol.7 No.1, 15–28.

Aziz, A., & Permana, U. (2015). Improvement of Performance Intrusion Detection System (Ids) Using Artificial Neural Network Ensemble, *80*(2), 191–202.

Balakrishnan, S., K, V., & a, K. (2014). Intrusion Detection System Using Feature Selection and Classification Technique. *International Journal of Computer Science and Application*, *3*(4), 145. https://doi.org/10.14355/ijcsa.2014.0304.02

Barber, D. (2010). Bayesian Reasoning and Machine Learning.*web4.cs.ucl.ac.uk/staff/D.Barber/textbook/090310.pdf*

Barot, V., Singh Chauhan, S., & Patel, B. (2014). Feature Selection for Modeling Intrusion Detection. *International Journal of Computer Network and Information Security*, *6*(7), 56–62. https://doi.org/10.5815/ijcnis.2014.07.08

Baykara, M., & Das, R. (2015). a Survey on Honeypot Technologies Used in Intrusion, (November), 14–19.International Journal of Computer Networks and Applications (IJCNA) Volume 2, Issue 5, September – October (2015)

Ben Mustapha, Y. (2015). Alert Correlation Towards an Efficient Response Decision Support.

Biswas, N. A., Shah, F. M., Tammi, W. M., & Chakraborty, S. (2016). FP-ANK: An improvised intrusion detection system with hybridization of neural network and K-means clustering over feature selection by PCA. *2015 18th International Conference on Computer and Information Technology, ICCIT 2015*, 317–322. https://doi.org/10.1109/ICCITechn.2015.7488089

Bouckaert, R. R. (2008). Bayesian Network Classifiers in Weka for Version 3-5-7. Retrieved from https://www.cs.waikato.ac.nz/~remco/weka.bn.pdf

Breetha, S., & Kavinila, R. (2013). Hierarchical Clustering For Cancer Discovery Using Range Check And Delta Check, iinternational Journal of Scientific and Research Publications, Volume 3, Issue 4, April 20131ISSN 2250-3153*3*(4), 2–5.

Chahar, V., Chhikara, R., Gigras, Y., & Singh, L. (2017). Significance of Hybrid Feature Selection Technique for Intrusion Detection Systems. *Indian Journal of Science and Technology*, *9*(48). https://doi.org/10.17485/ijst/2016/v9i48/105827

Chahira, J. M., Chuka, J. K. K., & Kemei, P. K. (2016). A Review of Intrusion Alerts Correlation Frameworks. *International Journal of Computer Applications Technology and Research*, *5*(4), 226–233. Retrieved from http://www.ijcat.com/archives/

volume5/issue4/ijcatr05041009.pdf

Chakir, E. M., Moughit, M., & Khamlichi, Y. I. (2017). An Effecient Method for Evaluating Alerts of Intrusion Detection Systems National School of Applied Sciences USMBA.

Chakraborty, N. (2013). Intrusion Detection System And Intrusion Prevention System: A Comparative Study. *International Journal of Computing and Business Research (IJCBR)*, *4*(2), 8. Retrieved from http://researchmanuscripts.com/May 2013/1.pdf%5 Cnhttp://www.researchmanuscripts.com/May2013/1.pdf

Chalak, A. (2011). Data Mining Techniques for Intrusion Detection and Prevention System, IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.8, *11*(8), 200–203.

Chand, N., Mishra, P., Krishna, C. R., Pilli, E. S., & Govil, M. C. (2017). A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection A Comparative Analysis of SVM and its Stacking with other Classification Algorithm for Intrusion Detection, (November). https://doi.org/10.1109/ ICACCA.2016.7578859

Chatur, A. D. P. N. (2014). Comparison of Firewall and Intrusion Detection System, *5*(1), 674–678. Retrieved from www.ijcsit.com/docs/Volume 5/vol5issue01/ijcsit 20140501145.pdf

Chaudhari, B., & Parikh, M. (2012). A Comparative Study of clustering algorithms Using weka tools. *Information Technology Journal*, *1*(2), 154–158. https://doi.org/10.3923/ itj.2006.551.559

Chaurasia, S., & Jain, A. (2014). Ensemble Neural Network and K-NN Classifiers for Intrusion Detection, International Journal of Computer Science and Information Technologies, Vol. 5 (2), 2014, 2481-2485*5*(2), 2481–2485.

Czarnowski, I., & Piotr, J. (2016). An Approach to Machine Classification Based on Stacked Generalization and Instance Selection.https://ieeexplore.ieee. org/iel7/7830913/ 7844217/07844994.pdf

Das, V., Pathak, V., Sharma, S., & Srikanth, M. (2010). Network Intrusion Detection System Based On Machine Learning, *International Journal Of Computer Science & Information Technology (IJCSIT), Vol 2, No 6,*.

Datasets, K., Mogal, D. G., Ghungrad, S. R., & Bhusare, B. B. (2017). NIDS using Machine Learning Classifiers on, *6*(4), 533–537. https://doi.org/10.17148/IJARCCE.2017.64102

Dewa, Z., & Maglaras, L. A. (2016). Data Mining and Intrusion Detection Systems. *International Journal of Advanced Computer Science and Applications*, *1*(1), 1:7. https://doi.org/10.1109/INFRKM.2010.5466919

Dhanabal, L., & Shantharajah, S. P. (2015). A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, *4*(6), 446–452. https://doi.org/10.17148/IJARCCE.2015.4696

Djemaa, B., & Okba, K. (2012). Intrusion Detection System : Hybrid Approach based Mobile Agent, 0–5.

Dubb Shruti, & Sood Yamini. (2013). Feature Selection Approach for Intrusion Detection System. *International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE)*, *2*(5), 47–53. Retrieved from http://warse.org/pdfs/2013 /icceitsp09.pdf

Duque, S., & Nizam, M. (2015). Using Data Mining Algorithms for Developing a Model for Intrusion Detection System (IDS). *Procedia - Procedia Computer Science*, *61*, 46–51. https://doi.org/10.1016/j.procs.2015.09.145

Elshoush, H. T. I. (2014a). An innovative framework for collaborative intrusion alert correlation. *Proceedings of 2014 Science and Information Conference, SAI 2014*, 607–614. https://doi.org/10.1109/SAI.2014.6918249

Elshoush, H. T. I. (2014b). Reducing the Correlation Processing Time by Using a Novel Intrusion Alert Correlation Model. *International Journal of Advanced Computer Science and Applications*, *Special Is*(Extended P). https://doi.org/10.14569/ SpecialIssue. 2014.040315

Elshoush, H. T., & Osman, I. M. (2012). An Improved Framework for Intrusion Alert, Proceedings of the World Congress on Engineering 2012 Vol I WCE 2012, July 4 - 6, 2012, London, U.K.

Fanfara, P., Dufala, M., & Radušovský, J. (2013). Autonomous hybrid honeypot as the future of distributed computer systems security. *Acta Polytechnica Hungarica*, *10*(6), 25–42.

Farhan, H. a., Naoum, R. S., & Islim, E. F. (2012). Intrusion detection model inspired by immune using K-means AND NAIVE BAYES AS A HYBRID LEARNING APPROACH. *International Journal of Academic Research*, *4*(4), 152–157. https://doi.org/10.7813/2075-4124.2012/4-4/A.22

Farid, D., Harbi, N., Ahmmed, S., Rahman, Z., & Rahman, C. M. (2010). Mining Network Data for Intrusion Detection through Naïve Bayesian with Clustering, International Journal of Computer and Information Engineering Vol:4, No:6.

Gambo, M. K., & Yasin, A. (2017). Hybrid Approach for Intrusion Detection Model Using Combination of K-Means Clustering Algorithm and Random Forest Classification. *Ijes*, *6*(1), 93–97.

Gervais, H., Munif, A., & Ahmad, T. (2016). Using QualityThreshold Distance to Detect Intrusion in TCP / IP Network. *2016 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, 80–84.

Gorton, D. a N. (2003). Extending Intrusion Detection with Alert Correlation and Intrusion Tolerance. *MPhil Thesis , Chalmers University of Technology*.

Govindarajan, M. (2014). Hybrid Intrusion Detection Using Ensemble of Classification Methods. *International Journal of Computer Network and Information Security*, *6*(2), 45–53. https://doi.org/10.5815/ijcnis.2014.02.07

Govindarajan, M. (2016). Evaluation of Ensemble Classifiers for Intrusion Detection, *10*(6), 1045–1053.Proceedings of the World Congress on Engineering and Computer Science 2012 Vol I WCECS 2012, October 24-26, 2012, San Francisco, USA

Guha, S. (2016). Attack Detection for Cyber Systems and Probabilistic State Estimation in Partially Observable Cyber Environments, *https://www.semanticscholar.org/.../Attack-*

Detection-for-Cyber-.

Haddadi, F., Khanchi, S., Shetabi, M., & Derhami, V. (2010). Intrusion Detection and Attack Classification Using Feed-Forward Neural Network. *Computer and Network Technology (ICCNT), 2010 Second International Conference On*, *3*(3), 1112–1115. https://doi.org/10.1109/ICCNT.2010.28

Hajamydeen, A. I., Udzir, N. I., Mahmod, R., & Abdul Ghani, A. A. (2016). An unsupervised heterogeneous log-based framework for anomaly detection. *Turkish Journal of Electrical Engineering and Computer Sciences*, *24*(3), 1117–1134. https://doi.org/ 10.3906/elk-1302-19

Harish, B. S., & Kumar, S. V. A. (2017). Anomaly based Intrusion Detection using Modified Fuzzy Clustering, *International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 4, Nº6*

Hasan, M. A. M., Nasser, M., Ahmad, S., & Molla, K. I. (2016). Feature Selection for Intrusion Detection Using Random Forest. *Journal of Information Security*, *7*(03), 129. https://doi.org/10.4236/jis.2016.73009

Heba, F. E., Darwish, A., Hassanien, A. E., & Abraham, A. (2010). Principle components analysis and Support Vector Machine based Intrusion Detection System. *2010 10th International Conference on Intelligent Systems Design and Applications*, 363–367. https://doi.org/10.1109/ISDA.2010.5687239

Hofmann, A., & Sick, B. (2011). Online intrusion alert aggregation with generative data stream modeling. *IEEE Transactions on Dependable and Secure Computing*, *8*(2), 282–294. https://doi.org/10.1109/TDSC.2009.36

Hussain, J., & Lalmuanawma, S. (2016). Feature Analysis, Evaluation and Comparisons of Classification Algorithms Based on Noisy Intrusion Dataset. *Procedia Computer Science*, *92,* 188–198. https://doi.org/10.1016/j.procs.2016.07.345

Hussain, J., Lalmuanawma, S., & Chhakchhuak, L. (2015). A Novel Network Intrusion Detection System using Two-stage Hybrid Classification Technique, International Journal of Computer & Communication Engineering Research (IJCCER)Volume 3-Issue 2.

Madbouly, A., M. Gody, A., & M. Barakat, T. (2014). Relevant Feature Selection Model Using Data Mining for Intrusion Detection System. *International Journal of Engineering Trends and Technology*, *9*(10), 501–512. https://doi.org/10.14445/22315381/IJETT-V9P296

Iafarov, R., Gad, R., & Kappes, M. (2015). Improving Attack Mitigation with a Cost-sensitive and Adaptive Intrusion Response System. *ICN : The Fourteenth International Conference on Networks*, (c), 134–139.

Ikram, S. T., & Cherukuri, A. K. (2016). Improving Accuracy of Intrusion Detection Model Using PCA and Optimized SVM, journal of Computing and Information Technology, Vol. 24,

Ingre, B., & Yadav, A. (2015). Performance analysis of NSL-KDD dataset using ANN. *International Conference on Signal Processing and Communication Engineering Systems - Proceedings of SPACES 2015, in Association with IEEE*, (January), 92–96. https://doi.org/10.1109/SPACES.2015.7058223

Jain, A., & Rana, J. L. (2016). Classifier Selection Models for Intrusion Detection System (Ids). *Informatics Engineering, an International Journal (IEIJ)*, *4*(1), 1–11. https://doi.org/ 10.5121/ieij.2016.4101

Jama, A. Y., Siraj, M., & Kadir, R. (2014). Towards Metamodel - based Approach for Information Security Awareness Management. *2014 International Symposium on Biometrics and Security Technologies (ISBAST)*, 316–321.

Janarthanan, T. (2017). Feature Selection in UNSW-NB15 and KDDCUP ' 99 datasets.http://shura.shu.ac.uk/15662

Joshi, N., & Srivastava, S. (2014). Improving Classification Accuracy Using Ensemble Learning Technique (Using Different Decision Trees), International Journal of Computer Science and Mobile Computing, Vol.3 Issue. 5,

Journal, I., Technological, F., Patel, A., & Tiwari, R. (2014). Bagging Ensemble Technique for Intrusion Detection, International Journal For Technological Research In EngineeringVolume 2, Issue 4

Juanchaiyaphum, J., Arch-int, N., & Arch-int, S. (2015). A Novel Lightweight Hybrid Intrusion Detection Method Using a Combination of Data Mining Techniques. *International Journal of Security and Its Applications*, *9*(4), 91–106. https:// doi.org/10.14257/ ijsia.2015.9.4.10

Kaliappan, J., Thiagarajan, R., & Sundararajan, K. (2015). Fusion of Heterogeneous Intrusion Detection Systems for Network Attack Detection, *Scientific World Journal Volume 2015, Article ID 314601, 8 pages http://dx.doi.org/10.1155/2015/314601*.

Kamesh, & Sakthi Priya, N. (2014). Security enhancement of authenticated RFID generation. *International Journal of Applied Engineering Research*, *9*(22), 5968–5974. https://doi.org/10.1002/sec

Kang, S. (2015). A Feature Selection Algorithm to Find Optimal Feature Subsets for Detecting DoS Attacks,

Kansra, M., & Chadha, P. D. (2016). Cluster Based detection of Attack IDS using Data Mining, IJEDR | Volume4, Issue 3| ISSN: 2321-9939.

Kaur, R., & Sachdeva, M. (2016). Analysis of Classification Approaches for Feature Selection in Intrusion Detection.International Journal of Advanced Research in An Empirical, *6*(9).

Kenaza, T., & Aiash, M. (2016). Toward an Efficient Ontology-based Event Correlation in SIEM. *Procedia - Procedia Computer Science*, *83*(Ant), 139–146. https://doi.org/ 10.1016/j.procs.2016.04.109

Khorshid, M. M. H., Abou-el-enien, T. H. M., & Soliman, G. M. A. (2015). A Comparison among Support Vector Machine and other Machine Learning Classification Algorithms,IPASJ International Journal of Computer Science (IIJCS).

Kidmose, E., Stevanovic, M., & Pedersen, J. M. (2016). Correlating intrusion detection alerts on bot malware infections using neural network. *2016 International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2016*. https://doi.org/10.1109/CyberSecPODS.2016.7502344

Kohavi, R., & Mateo, S. (1999). C5.1.5 Bayesian Classification, https://www.sciencedirect.com/ topics/.../bayesian-classification1.

Kovac, S. (2012). Suitability analysis of data mining tools and methods, https://is.muni.cz/th/.../suitability_analysis_of_data_mining_tools.p.

Kulkarni, R. D. (2014). Using Ensemble Methods for Improving Classification of the KDD CUP ’ 99 Data Set, Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 16, Issue 5, Ver. II.

Kumar, G., & Kumar, K. (2012). The Use of Multi-Objective Genetic Algorithm Based Approach to Create Ensemble of ANN for Intrusion Detection. *International Journal of Intelligence Science*, *2*(October), 115–127.

Kumar, K. (2016). Network Intrusion Detection with Feature Selection Techniques using Machine-Learning Algorithms, Iinternational Journal of Computer Applications (0975 – 8887) Volume 150 –No.12,

Kumar, S., & Naveen, D. C. (2016). A Survey on Improving Classification Performance Using Data Pre processing And Machine Learning Methods on NSL-KDD Data, *05*(16156), 16156–16161. https://doi.org/10.18535/ijecs/v5i4.17

Kumar, V., Chauhan, H., & Panwar, D. (2013). K-Means Clustering Approach to Analyze NSL-KDD Intrusion Detection Dataset, International Journal of Soft Computing and Engineering (IJSCE)ISSN: 2231-2307, Volume-3, Issue-4.

Kumar, Y. (2016). AI based Hybrid Ensemble Technique for Network Security, International Journal of Computer Applications (0975 –8887) International Conference on Advancesin Emerging Technology (ICAET 2016).

Kuźniar, K., & Zając, M. (2015). Some methods of pre-processing input data for neural networks, *https://www.vantagepointsoftware.com/.../preprocessing-data-neur*..

Lakshmi, S. V., Prabakaran, T. E., & Ph, D. (2014). Application of k-Nearest Neighbour Classification Method for Intrusion Detection in Network Data, International Journal of Computer Applications (0975 –8887) Volume 97–No.7, July 2014.

Li, G., & Yan, Z. (2018). Data Fusion for Network Intrusion Detection : A Review, Security and Communication Networks Volume 2018, Article ID 8210614, https://doi.org/10.1155/2018/8210614.

Li, L., Yu, Y., Bai, S., Cheng, J., & Chen, X. (2018). Towards Effective Network Intrusion Detection : A Hybrid Model Integrating Gini Index and GBDT with PSO, *Journal of Sensors, Volume 2018, Article ID 1578314, https://doi.org/10.1155/2018/1578314*

Long, L., Wang, X., & Zhu, X. (2015). Machine Learning in Network Intrusion Detection, Int'l Conf. Security and Management | SAM'16 |.

Madbouly, A. I. (2016). Enhanced relevant feature selection model for intrusion detection systems Tamer M . Barakat, Int. J. Intelligent Engineering Informatics, Vol. 4, No. 1, 2016.

Mahmood, D. Y., & Hussein, M. A. (2013). Intrusion Detection System Based on K-Star Classifier and Feature Set Reduction. *International Organization of Scientific Research Journal of Computer Engineering (IOSR-JCE) Vol*, *15*(5), 107–112.

Mallissery, S., Kolekar, S., & Ganiga, R. (2014). Accuracy Analysis of Machine Learning Algorithms for Intrusion Detection System using NSL-KDD Dataset, https://www.semanticscholar.org/.../Accuracy-Analysis-of-Machine.

Manandhar, P. (2014). A Practical Approach to Anomaly - based Intrusion Detection System by Outlier Mining in Network Traffic, Retrieved from *https://www.ccsl.carleton.ca/mmcna. old/publications/NS0907.pdf*

Miškovic, V. (2014). Machine Learning of Hybrid Classification Models for Decision Support. *Proceedings of the 1st International Scientific Conference - Sinteza 2014*, 318–323. Retrieved from https://doi.org/10.15308/sinteza-2014-318-323

Moustafa, N. (2015). A hybrid feature selection for network intrusion detection systems : Central points Pproceeding of the 16th Australian Information Warfare Conference (pp. 5-13), held on the 30 November - 2 December,2015, Edith Cowan University, Joon dalup Campus, Perth, Western Australia.

Moustafa, N., & Slay, J. (2015). The significant features of the UNSW-NB15 and the KDD99 Data sets for Network Intrusion Detection Systems, 44th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)

Muchammad, K. (2015). Detecting Intrusion Using Recursive Clustering and Sum of Log Distance to Sub-centroid. *Procedia - Procedia Computer Science*, *72*, 446–452. https://doi.org/10.1016/j.procs.2015.12.125

Mukkamala, S., Janoski, G., & Sung, A. (1998). Intrusion Detection : Support Vector Machines and Neural Networks.

Mukosera, M., Mpofu, T. P., & Masaiti, B. (2014). Analysis of NSL-KDD Dataset for Fuzzy Based Intrusion Detection System, *3*(6), 2012–2015.

Nachenberg, C. (2011). A Window Into Mobile Device Security, 23. Retrieved from http://digcert.com/docs/symantec/b-mobile-device-security-exec-summary_WP.en-us.pdf

Nadiammai, G. V., & Hemalatha, M. (2014). Effective approach toward Intrusion Detection System using data mining techniques. *Egyptian Informatics Journal*, *15*(1), 37–50. https://doi.org/10.1016/j.eij.2013.10.003

Nagle, M. K., & Chaturvedi, S. K. (2013). Feature Extraction Based Classification Technique for Intrusion Detection System, Proceedings of the International MultiConference of Engineers and Computer Scientists 2015 Vol I, IMECS 2015, March 18 - 20, 2015, Hong Kong.

Nalavade, K. (2014). Evaluation of K-Means Clustering for Effective Intrusion Detection and Prevention in Massive Network Traffic Data, International Journal of Computer Applications (0975 –8887) Volume 96–No.7,

Nevlud, P., Bures, M., Kapicak, L., & Zdralek, J. (2013). Anomaly-based Network Intrusion Detection Methods Keywords Detection of Network Anomalies, 468–474. https://doi.org/10.15598/aeee.v11i6.877

Noureldien, N. A., & Yousif, I. M. (2016). Accuracy of Machine Learning Algorithms in

Detecting DoS Attacks Types, *6*(4), 89–92. https://doi.org/10.5923/j.scit.20160604.01

Othman, M., & Maklumat, T. (1999). Mobile Computing and Communications: An Introduction. *Malaysian Journal of Computer …*, *12*(2), 71–78. Retrieved from http://icmsm2009.um.edu.my/filebank/published_article/1750/74.pdf

Pahwa, S. (2016). Support Vector Machine with Artificial Neural Network Using Integer Datasets,International Journal of Advanced Research in Comparative Study of, *6*(11), 200–205.

Palanisamy, S. (2006). Association rule based classification, (May), 74. Retrieved from http://www.wpi.edu/Pubs/ETD/Available/etd-050306-131517/

Panda, M., Abraham, A., & Patra, M. R. (2012). A hybrid intelligent approach for network intrusion detection. *Procedia Engineering*, *30*(2011), 1–9. https://doi.org/10.1016 /j.proeng.2012.01.827

Panda, M., Abraham, A., & Patra, M. R. (2015). Hybrid intelligent systems for detecting network intrusions, (July 2012), 2741–2749. https://doi.org/10.1002/sec

Panwar, S. S. (2014). of Computer © I a E M E Data Reduction Techniques To Analyze Nsl-Kdd Dataset, 21–31.

Parsaei, M. R., Rostami, S. M., & Javidan, R. (2016). A Hybrid Data Mining Approach for Intrusion Detection on Imbalanced NSL-KDD Dataset, *7*(6), 20–25.

Perez, D., Astor, M. A., Abreu, D. P., & Scalise, E. (2017). Intrusion Detection in Computer Networks Using Hybrid Machine Learning Techniques.

Pradhan, R. (2014). Performance Assessment of Robust Ensemble Model for Intrusion Detection using Decision Tree Techniques, *3*(3), 78–86.

Pundir, S. L., & Amrita. (2013). Feature Selection Using Random Forest in Intrusion Detection. *International Journal of Advances in Engineering & Technology*, *6*(3), 1319–1324.

Rahayu, S. S., Robiah, Y., Shahrin, S., Zaki, M. M., Faizal, M. A., & Zaheera, Z. A. (2010). Advanced Trace Pattern For Computer Intrusion Discovery, *2*(6), 200–207. Retrieved from http://arxiv.org/abs/1006.4569

Ramaki, A. A., Khosravi-Farmad, M., & Bafghi, A. G. (2015). Real time alert correlation and prediction using Bayesian networks. *2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)*, *978*, 98–103. https://doi.org/10.1109/ISCISC.2015.7387905

Rasmi, M., & Al Qerem, A. (2015). PNFEA: A Proposal Approach for Proactive Network Forensics Evidence Analysis to Resolve Cyber Crimes. *International Journal of Computer Network and Information Security*, *7*(2), 25–32. https://doi.org/10.5815/ijcnis.2015.02.03

Rathore, D., & Jain, A. (2012). Design Hybrid method for intrusion detection using Ensemble cluster classification and SOM network, (3), 3–8.

Revathi, M. (2000). Network Intrusion Detection System Using Reduced. *Journal of Computer Science*, *2*(1), 61–67. https://doi.org/10.1146/annurev.physiol.62.1.939

Roschke, S., Cheng, F., & Meinel, C. (2010). A flexible and efficient alert correlation platform for distributed IDS. *Proceedings - 2010 4th International Conference on Network and System Security, NSS 2010*, 24–31. https://doi.org/10.1109/NSS.2010.26

Sahu, P., & Miri, R. (2017). A Hybrid Technique for creating classification, *5*(Vi), 82–84.

Sampat, R., & Sonawani, S. (2015). Network Intrusion Detection Using Dynamic Fuzzy C Means Clustering ., *2*(October), 135–141.

Sampath, U., Perera, K., & Thanthrige, M. (2016). Hidden Markov Model Based Intrusion Alert Prediction, (September), 1–8.

Sannady, V., & Gupta, P. (2016). Intrusion Detection Model in Data Mining Based on Ensemble Approach, 1654–1658.

Scholar, M. T. (2017). An Efficient NIDS by using Hybrid Classifiers Decision Tree & Decision Rules, *2*(1), 76–79.

Science, C. (2015). A Hybrid Approach to improve the Anomaly Detection Rate Using Data Mining Techniques, (July).

Sendi, A. S., Dagenais, M., Jabbarifar, M., & Couture, M. (2012). Real time intrusion prediction based on optimized alerts with Hidden Markov model. *Journal of Networks*, *7*(2), 311–321. https://doi.org/10.4304/jnw.7.2.311-321

Sendi, A. S., Jabbarifar, M., Shajari, M., & Dagenais, M. (2010). FEMRA: Fuzzy Expert Model for Risk Assessment. *2010 Fifth International Conference on Internet Monitoring and Protection*, 48–53. https://doi.org/10.1109/ICIMP.2010.15

Sesmero, M. P., Ledezma, A. I., & Sanchis, A. (2015). Generating ensembles of heterogeneous classifiers using, *5*(February), 21–34. https://doi.org/10.1002/widm.1143

Shah, R. A., Qian, Y., & Mahdi, G. (2017). Group Feature Selection via Structural Sparse Logistic Regression for IDS. *Proceedings - 18th IEEE International Conference on High Performance Computing and Communications, 14th IEEE International Conference on Smart City and 2nd IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2016*, 594–600. https://doi.org/10.1109/HPCC-SmartCity-DSS.2016.0089

Shahadat, N., Hossain, I., Rohman, A., & Matin, N. (2017). Experimental Analysis of Data Mining Application for Intrusion Detection with Feature reduction, 209–216.

Shahbaz, M. B., Wang, X., Behnad, A., & Samarabandu, J. (2016). On Efficiency Enhancement of the Correlation-based Feature Selection for Intrusion Detection Systems. https://doi.org/10.1109/IEMCON.2016.7746286

Shameli-Sendi, A., & Dagenais, M. (2014). ARITO: Cyber-attack response system using accurate risk impact tolerance. *International Journal of Information Security*, *13*(4), 367–390. https://doi.org/10.1007/s10207-013-0222-9

Shameli-Sendi, A., & Dagenais, M. (2015). ORCEF: Online response cost evaluation framework for intrusion response system. *Journal of Network and Computer Applications*, *55*, 89–107. https://doi.org/10.1016/j.jnca.2015.05.004

Shameli-Sendi, A., Desfossez, J., Dagenais, M., & Jabbarifar, M. (2013). A retroactive-burst framework for automated intrusion response system. *Journal of Computer Networks and Communications*, *2013*. https://doi.org/10.1155/2013/134760

Shameli Sendi, A. (2013). System health monitoring and proactive response activation. *ProQuest Dissertations and Theses*, *NR95250*, 180. Retrieved from http://ezproxy.net.ucf.edu/login?url=http://search.proquest.com/docview/1460765120?accountid=10003%5Cnhttp://sfx.fcla.edu/ucf?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:dissertation&genre=dissertations+&+theses&sid=ProQ:ProQuest+Dissertations+&+

Sharma, M. (2016). INTRUSION DETECTION SYSTEM BY USING SVM & ANN, *6*(2), 149–155.

Sharma, N., Bajpai, A., & Litoriya, R. (2012). Comparison the various clustering algorithms of weka tools. *International Journal of Emerging Technology and Advanced Engineering*, *2*(5), 73–80.

Sharma, R., Shrivastava, M., Lnct, P. G. S., & Bhopal, M. P. (2012). Intrusion Awareness Based on Data Fusion and SVM Classification, *2*(2).

Shen, Y., Yu, F., Zhang, L.-F., An, J.-Y., & Zhu, M.-L. (2005). An intrusion detection system based on system call. *First IEEE and IFIP International Conference in Central Asia on Internet, 2005*, *2005*(1), 21–25. https://doi.org/10.1109/CANET.2005.1598184

Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A Deep Learning Approach to Network Intrusion Detection, *2*(1), 41–50.

Shrivastava, A., Baghel, M., & Gupta, H. (2013). A Review of Intrusion Detection Technique by Soft Computing and Data Mining Approach, (3), 224–228.

Singh, A. (2009). Improving information security risk management. Retrieved from http://conservancy.umn.edu/handle/11299/58967%5Cnhttp://conservancy.umn.edu/bitstream/11299/58967/1/Singh_umn_0130E_10820.pdf

Singh Arora, I., Kaur Bhatia, G., & Professor, A. (2016). Comparative Analysis of Classification Algorithms on KDD " 99 Data Set. *I.J. Computer Network and Information Security*, *9*(9), 34–40. https://doi.org/10.5815/ijcnis.2016.09.05

Singh, D. (2013). Intrusion Detection System based on Fuzzy C Means Clustering and Probabilistic Neural Network, *74*(2), 30–33.

Singh, H., & Kumar, D. (2015). A study on Performance analysis of various feature selection techniques in intrusion detection system. *International Journal of Advance Research in Computer Science and Management Studies*, *3*(6), 2321–7782.

Siqueira, I. G., Ruiz, L. B., & Loureiro, A. a. F. (2014). Coverage area management for wireless sensor networks. *Internation Journal of Network Management*, (October 2005), 17–31. https://doi.org/10.1002/nem

Siraj, M. M., & Hashim, S. Z. M. (2008). Network intrusion alert correlation challenges and techniques. *Journal of Teknologi Maklumat*, *20*(Dlsember 2008), 12–36.

Siraj, M. M., Hussein, H., Albasheer, T., & Din, M. M. (2015). Towards Predictive Real-time Multi-sensors Intrusion Alert Correlation Framework. *Indian Journal of Science and Technology ISSN*, *8*(12), 974–6846. https://doi.org/10.17485/ijst/2015/v8i12/70658

Siraj, M. M., Maarof, M. A., & Hashim, S. Z. M. (2009a). A Hybrid Intelligent Approach for Automated Alert Clustering and Filtering in Intrusion Alert Analysis. *International Journal of Computer Theory and Engineering*, *1*(5), 539–545. https://doi.org/10.7763/IJCTE.2009.V1.87

Siraj, M. M., Maarof, M. A., & Hashim, S. Z. M. (2009b). Intelligent alert clustering model for network intrusion analysis. *International Journal of Advances in Soft Computing and Its Applications*, *1*(1), 33–48.

Siraj, M. M., Maarof, M. A., & Hashim, S. Z. M. (2009c). Intelligent clustering with PCA and unsupervised learning algorithm in intrusion alert correlation. *5th International Conference on Information Assurance and Security, IAS 2009*, *1*, 679–682. https://doi.org/10.1109/IAS.2009.261

Siraj, M., Maarof, M. A., Zaiton, S., & Hashim, M. (2011). Network Intrusion Alert Aggregation Based on PCA and Expectation Maximization Clustering Algorithm, *2*, 395–399.

Siraj, M., Maarof, M. A., Zaiton, S., Hashim, M., Din, M. M., & Kadir, R. (2010). Integration of PCA and Levenberg-Marquardt Neural Network in Alert Correlation, (Icinc), 133–137.

Sivakumar, B., & Srilatha, K. (2016). A novel method to segment blood vessels and optic disc in the fundus retinal images. *Research Journal of Pharmaceutical, Biological and Chemical Sciences*, *7*(3), 365–373. https://doi.org/10.15680/IJIRCCE.2016.

Smith, R., Japkowicz, N., & Dondo, M. (2008). Using unsupervised learning for network alert correlation. *Advances in Artificial*. Retrieved from http://www.springerlink.com/index/29p1682l1l484508.pdf

Solanki, M. (2014). Intrusion Detection System by using K-Means clustering , C 4 . 5 , FNN , SVM classifier, *3*(6).

Song, J. (2016). Feature Selection for Intrusion Detection System Jingping Song Declaration and Statement, Ph.D. Thesis Department of Computer Science Institute of Mathematics, Physics and Computer Science Aberystwyth University.

Subbulakshmi, T., Mathew, G., & Shalinie, M. (2010). Real Time Classification and Clustering of Ids Alerts Using Machine Learning Algorithms. *International Journal of Artificial Intelligence & Applications (IJAIA)*, *1*(1), 1–9.

Sunita, S., Chandrakanta, B. J., & Chinmayee, R. (2016). A Hybrid Approach of Intrusion Detection using ANN and FCM, *3*(2), 6–14.

Syarif, I., Prugel-Bennett, A., & Wills, G. (2012). Unsupervised clustering approach for network anomaly detection. *Networked Digital Technologies*, *293*. https://doi.org/ 10.1007/ 978-3-642-30507-8

Thaseen, S., & Kumar, A. (2017). Intrusion detection model using fusion of chi-square feature selection and multi class SVM. *Journal of King Saud University - Computer and Information Sciences*, *29*(4), 462–472. https://doi.org/10.1016/j.jksuci.2015.12.004

Thaseen, S., & Kumar, C. A. (2013). An analysis of supervised tree based classifiers for intrusion detection system. *Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, PRIME 2013*, (February 2013), 294–299. https://doi.org/10.1109/ICPRIME.2013.6496489

Thaseen, S., & Kumar, C. A. (2016). Intrusion Detection Model using PCA and Ensemble of Classifiers, *16*(2), 15–38.

Thomas, C., & Balakrishnan, N. (2008). Performance enhancement of Intrusion Detection Systems using advances in sensor fusion. *2008 11th International Conference on Information Fusion*, 1–7. https://doi.org/10.1109/ICIF.2008.4632412

Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications*, *36*(10), 11994–12000. https://doi.org/ 10.1016/j.eswa.2009.05.029

Urvashi, M., & Jain, M. A. (2015). A survey of IDS classification using KDD CUP 99 dataset in WEKA. *International Journal of Scientific & Engineering Research*, *6*(11), 947–954. Retrieved from http://www.ijser.org

Valeur, F. (2006). Real-Time Intrusion Detection Alert Correlation. *Thesis for the degree of philosophiae doctorTrondheim, October 2010Norwegian University of Science and Technology Faculty of Information Technology, Mathematics and Electrical Engineering Department of Telematics*

Vaughn, R. B., & Bridges, S. M. (2004). Intrusion Sensor Data Fusion in an Intelligent Intrusion Detection System Architecture 1 Ambareen Siraj, *00*(C), 1–10.

Verma, P. (2016). Performance of Detection Attack using IDS Technique, *4*(3), 624–629.

Wahba, Y., ElSalamouny, E., & ElTaweel, G. (2015). Improving the Performance of Multi-class Intrusion Detection Systems using Feature Reduction. *Ijcsi*, *12*(3), 255–262. Retrieved from http://arxiv.org/abs/1507.06692

Wang, G., Hao, J., Ma, J., & Huang, L. (2010). Expert Systems with Applications A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. *EXPERT SYSTEMS WITH APPLICATIONS*. https://doi.org/10.1016/j.eswa.2010.02.102

Wang, W. (2010). A graph oriented approach for network forensic analysis, A dissertation submitted to the graduate faculty in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY, Iowa State University Ames, Iowa.

Witten, I. H., Frank, E., & Hall, M. a. (2011). *Data Mining: Practical Machine Learning Tools and Techniques (Google eBook). Complementary literature None*. https://doi.org/0120884070, 9780120884070

Wu, X., Kumar, V., Ross, Q. J., Ghosh, J., Yang, Q., Motoda, H., … Russell, I. (2006). BUsiness Intelligence and Aalytics : From Big Data To Big Impact. *Mis Quarterly*, *11*(1), 1–37. https://doi.org/10.1007/s10115-007-0114-2

Yang, Y. (2011). Tutorial on Classification Outline. *Masterchemoinfo.U-Strasbg.Fr*, 1–27. Retrieved from http://masterchemoinfo.u-strasbg.fr/Documents/ TutoChemo /classification.pdf

Yusof, R., Selamat, S. R., & Sahib, S. (2008). Intrusion Alert Correlation Technique Analysis for Heterogeneous Log. *IJCSNS International Journal of Computer Science and Network Security*, *8*(9), 132–138.

Yusof, R., Selamat, S. R., Sahib, S., Mas'ud, M. Z., & Abdollah, M. F. (2011). Enhanced alert correlation framework for heterogeneous log. *Communications in Computer and Information Science*, *251 CCIS*(PART 1), 107–122. https://doi.org/10.1007/978-3-642-25327-0_10

Zainal, A., Maarof, M. A., & Shamsuddin, S. M. (2009). Ensemble Classifiers for Network Intrusion Detection System, *PLoS ONE*, *11*(11), 1–18. https://doi.org/10.1371 /journal.pone.0166017.

Zhang, Y., Huang, S., Guo, S., & Zhu, J. (2011). Multi-sensor Data Fusion for Cyber Security Situation Awareness. *Procedia Environmental Sciences*, *10*(Esiat), 1029–1034. https://doi.org/10.1016/j.proenv.2011.09.165

Zhou, C. V., Leckie, C., & Karunasekera, S. (2010). A survey of coordinated attacks and collaborative intrusion detection. *Computers and Security*, *29*(1), 124–140. https://doi.org/10.1016/j.cose.2009.06.008

APPENDICES

APPENDIX I : Description of NSL KDD_Data Set

NO	Feature	Description	Type
1	Duration	Length of the connection.	Basic Features
2	protocol type	Connection protocol (e.g. tcp, udp)	Basic Features
3	Service	Destination service (e.g. telnet, ftp)	Basic Features
4	Flag.	Normal or error status of the connection	Basic Features
5	source byte	Bytes sent from source to destination	Basic Features
6	destination bytes	Bytes sent from destination to source	Basic Features
7	Land	1 - Connection is from/to the same host/port; 0 – otherwise.	Basic Features
8	Wrong Fragment	Number of “wrong” fragments	Basic Features
9	Urgent	Number of urgent packets	Content Features
10	Hot	number of “hot indicators”.	Content Features
12	num_failed_logins	number of failed login attempts	Content Features
13	logged in	1 - successfully logged in; 0 – otherwise	Content Features
14	num_compromise	number of “compromised” conditions.	Content Features
15	root shell	number of “compromised” conditions.	Content Features
16	su_attempted	1 - root shell is obtained; 0 – otherwise.	Content Features
17	num_root	number of “root” accesses.	Content Features
18	num_file_creations	number file creation operations	Content Features
19	num_shells	number of shell prompts	Content Features
20	Num_access_files	number of operations on access	Content Features

		control files	
21	Num_outbound_cmds	number of outbound commands in a ftp session	Content Features
22	is_hot_login	1 - the login belongs to the “hot” list; 0 – otherwise.	Content Features
23	is_guest_login	1 - the login is a “guest”login; 0 - otherwise	Content Features
24	Count	number of connections to the same host as the current connection in the past 2 seconds	Time-based Traffic Feature
25	srv_count	number of connections to the same service as the current connection in the past 2 seconds	Time-based Traffic Features
26	serror_rate	% of connections that have “SYN” error	Time-based Traffic Features
27	rerror_rate	% of connections that have “REJ” errors	Time-based Traffic
28	same srv rate	% of connections to the same service	Time-based Traffic
29	diff srv rate	% of connections to different services	Time-based Traffic
30	srv_serror_rate	% of connections that have “SYN” errors	Time-based Traffic
31	srv_rerror_rate	% of connections that have “REJ” errors	Time-based Traffic
32	srv_diff_host_rate	% of connections to different hosts	Time-based Traffic
33	Dst_host_count	count of connections having the same destination host	Host-based Traffic Feature
34	dst_host_srv_count	count of connections having the same destination host and using	Host-based Traffic Feature

		the same service	
35	dst_host_same_srv_ rate	% of connections having the same destination host and using the same service	Host-based Traffic Feature
36	dst_host_diff_srv_rate	% of different services on the current host	Host-based Traffic Feature
37	dst_host_same_src_port_rat	% of connections to the current host having the same src port	Host-based Traffic Feature
38	Dst_host_ srv_diff_host_rate	% of connections to the same service coming from different hosts	Host-based Traffic Feature
39	Dst_host_srv_rerror_rate	% of connections to the current host and specified service that have an S0 error	Host-based Traffic Feature
40	dst_host_serror_rate	% of connections to the current host that have an S0 error	Host-based Traffic Feature
41	dst_host_srv_serror_rate	% of connections to the current host and specified service that have an S0 error	Host-based Traffic Feature

APPENDIX II: Description of USW-NB15_Features

NO	NAME	TYPE	DESCRIPTION
1	Srcip	Nominal	Source IP address
2	Sport	Integer	Source port number
3	Dstip	Nominal	Destination IP address
4	Dsport	Integer	Destination port number
5	Proto	Nominal	Transaction protocol
6	State	Nominal	Indicates to the state and its dependent protocol
7	Dur	Float	Record total duration
9	dbytes	Integer	Destination to source transaction bytes
10	Sttl	Integer	Source to destination time to live value
11	Dttl	Integer	Destination to source time to live value
12	Sloss	Integer	Source packets retransmitted or dropped
13	Dloss	Integer	Destination packets retransmitted or dropped
14	service	Nominal	http, ftp, smtp, ssh, dns, ftp-data ,irc and (-) if not much used service
15	Sload	Float	Source bits per second
16	Dload	Float	Destination bits per second
17	Spkts	Integer	Source to destination packet count
18	Dpkts	Integer	Destination to source packet count
19	swin	Integer	Source TCP window advertisement value

20	dwin	Integer	Destination TCP window advertisement value
21	stcpb	Integer	Source TCP base sequence number
22	dtcpb	Integer	Destination TCP base sequence number
23	smeansz	Integer	Mean of the ?ow packet size transmitted by the src
24	dmeansz	Integer	Mean of the ?ow packet size transmitted by the dst
25	trans_depth	Integer	Represents the pipelined depth into the connection of http request/response transaction
26	res_bdy_len	Integer	Actual uncompressed content size of the data transferred from the server’s http service.
27	Sjit	Float	Source jitter (mSec)
28	Djit	Float	Destination jitter (mSec)
29	Stime	Timestamp	record start time
30	Ltime	Timestamp	record last time
31	Sintpkt	Float	Source interpacket arrival time (mSec)
32	Dintpkt	Float	Destination interpacket arrival time (mSec)
33	tcprtt	Float	TCP connection setup round-trip time, the sum of ’synack’ and ’ackdat’.
34	synack	Float	TCP connection setup time, the time between the SYN and the SYN_ACK packets.
35	ackdat	Float	TCP connection setup time, the time between the SYN_ACK and the ACK packets.
36	is_sm_ips_ports	Binary	If source (1) and destination (3)IP addresses equal and port numbers (2)(4) equal then, this variable takes value 1 else 0

37	ct_state_ttl	Integer	No. for each state (6) according to specific range of values for source/destination time to live (10) (11).
38	ct_flw_http_mthd	Integer	No. of flows that has methods such as Get and Post in http service.
39	is_ftp_login	Binary	If the ftp session is accessed by user and password, then 1 else 0.
40	ct_ftp_cmd	Integer	No of flows that has a command in ftp session.
41	ct_srv_src	Integer	No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26).
42	ct_srv_dst	Integer	No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26).
43	ct_dst_ltm	Integer	No. of connections of the same destination address (3) in 100 connections according to the last time (26).
44	ct_src_ ltm	Integer	No. of connections of the same source address (1) in 100 connections according to the last time (26).
45	ct_src_dport_ltm	Integer	No of connections of the same source addres (1) and the destination port (4) in 100 connections according to the last time (26).
46	ct_dst_sport_ltm	Integer	No of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26).
47	ct_dst_src_ltm	Integer	No of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26).

48	attack_cat	Nominal	The name of each attack category. In this data set , nine categories e.g. Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms
49	Label	Binary	0 for normal and 1 for attack records

APPENDIX III: Algorithm: Best Features Selection

```
Input: Datasets with Common Reduced Features
Output: A set of most relevant features
/*Stage 4.1: Gradually Delete Phase*/
Starting from the common features set CS[i]
Rank the CS[i], U2R[i], R2L [i], PROBE[i], and DOS[i] based on
    The importance of the feature to the attack type (relevance value)
    How many attack type the feature can detect
    How many algorithms select this feature for each attack type
For j = 1 to i
    If a feature is (used to detect ONLY DOS) AND it is (in the lowest ranked list of DOS)
    Else if a feature is (used to detect ONLY PROBE) AND it is (in the lowest ranked list of PROBE)
    Else if a feature is (used to detect ONLY R2L) AND it is (in the lowest ranked list of R2L)
    Else if a feature is (used to detect ONLY U2R) AND it is (in the lowest ranked list of U2R)
    Else if a feature is (used to detect DOS and PROBE) AND it is (in the lowest ranked list of DOS and PROBE)
        Delete this feature
    Update the CS[j]
    Evaluate performance of the updated CS[j]
```

```
If better performance for U2R, R2L, and PROBE
    Confirm feature deletion
    Update CS[j]
    Update BSA
Else
    keep this feature
    Update CS[j]
    Update BSA
Next j
/*End of Gradually Delete Phase*/
/* Stage 4.2: Gradually Add Phase*/
Start by a common selected set CF(i) of features that are:
Selected as important for all attack types
Selected by all algorithms with high relevance value
Evaluate the performance of CF(i) Æ BSA
Do until Max BSA
    Add the top ranked feature form the U2R(j) set to CF(i)
    Evaluate the performance of CF(i)
    If performance > BSA
    Confirm adding this feature
    Update CF(i)
    Update U2R(j)
    Update BSA
Else
    Change the feature importance to lowest rank
Update U2R(j)
End if
Add the top ranked feature form the R2L(j) set to CF(i)
Evaluate the performance of CF(i)
If performance > BSA
    Confirm adding this feature
    Update CF(i)
Update R2L(j)
    Update BSA
Else
    Change the feature importance to lowest rank
    Update R2L(j)
End if
Add the top ranked feature form the PROBE(j) set to CF(i)
Evaluate the performance of CF(i)
If performance > BSA
    Confirm adding this feature
    Update CF(i)
    Update PROBE(j)
    Update BSA
Else
    Change the feature importance to lowest rank
    Update PROBE(j)
End if
Add the top ranked feature form the DOS(j) set to CF(i)
Evaluate the performance of CF(i)
If performance > BSA
    Confirm adding this feature
    Update CF(i)
    Update DOS(j)
    Update BSA
Else
    Change the feature importance to lowest rank
    Update DOS(j)
    End if
Repeat
Return BSA and CF(i)
/*End of Gradually Add Phase*/
```