NETWORK INTRUSION DETECTION

WITH

NAÏVE BAYES CLASSIFICATION AND SELF ORGANIZING MAPS

Master's Student: Mubeen Iqbal

Supervised by: A/Prof. Quang Ha

FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY (FEIT)

UNIVERSITY OF TECHNOLOGY SYDNEY (UTS)

August 2014

CERTIFICATE OF ORIGINAL AUTHORSHIP

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Mubeen Iqbal

Acknowledgment

First and foremost, I offer my sincerest gratitude to my advisor, **A/Prof. Quang Ha**, who has backed me, all around my thesis with his perseverance and knowledge. This research would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

Secondly, I would like to thank my research colleagues for their encouragement throughout my research work.

Finally, I would like to thank my parents for their unending love, support and understanding during my research work.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

IDS	Intrusion Detection System
LSHSN	Large-Scale High-Speed Networks
KDD	Knowledge Discovery in Data Competitions
ITS	Intrusion Tolerant System
NIDS	Network Intrusion Detection System
MADAM	Mining Audit Data for Automated Model
HIDS	Host Based Intrusion Detection System
IDES	Intrusion Detection Expert System
NIDES	Network Intrusion Detection Expert System
SVM	Support Vector Machines
SOM	Self Organizing Map
NB	Naïve Bayes
ML	Machine Learning
DARPA	Defense Advanced Research Projects Agency
HMM	Hidden Markov Model
HTML	Hyper Text Markup Language
DOS	Denial-of- Service
U to R	User to Root Attacks
R to L	Remote to Local Attacks
DR	Detection Rate
FPR	False Positive Rate
WEKA	Waikato Environment for Knowledge Analysis

Abstract

In this digital period, internet has turned into an indispensable wellspring of correspondence in just about every calling. With the expanded use of system engineering, its security has developed to be exceptionally discriminating issue as the workstations in distinctive association hold very private data and touchy information. The system used to screen the system security is known as Network detection. Intrusion detection is to get ambushes against a machine structure. It is a discriminating enhancement great to go part and additionally an element extent of examination. In Information Security, Intrusion recognizable proof is the showing of placing exercises that attempt to deal the protection, respectability or availability of a benefit. It accepts an astoundingly key part in waylay area, security check and framework inspect. One of the vital tests to Intrusion Detection is the issue of misjudgement, misdetection and unsuccessful deficiency of steady response to the strike. In the past years, as the second line of boundary after firewall, the Intrusion Detection strategy has got speedy progression.

This research work prepares two diverse Machine Learning techniques, both supervised and unsupervised, for Network Intrusion Detection. These techniques are Naïve Bayes (supervised learning) and Self Organizing Maps (unsupervised learning). The KDD Cup 99 dataset is utilized for Intrusion Detection Problem. As KDD Cup 99 dataset holds some symbolic attribute and also numeric attributes, two sorts of transformation technique have been utilized for these properties. These are conditional probabilities conversion technique and indicator variables transformation. The two machine learning procedures are prepared on both kind of transformed dataset and afterward their outcomes are looked at with respect to the correctness of intrusion detection.

Keywords: Network Intrusion Detection, supervised learning, unsupervised learning, Self-Organizing Map, Naïve Bayes, Conditional Probability Symbolic Conversion, Indicator Variable Symbolic Conversion, KDD Cup 1999.

CHAPTER 1
INTRODUCTION

This chapter portrays the Network Intrusion and its destructive consequences for the security of connection data. It likewise depicts the importance and need of Network Intrusion Detection Systems to address the issue of network intrusions. It demonstrates the methodology taken to execute Network Intrusion Detection System. The structure is arranged to cover a brief overview, followed by the introduction of network intrusion and its detection. The chapter also highlights the problem statement the thesis deals with and the objectives and scope of the thesis. Finally, a layout of the intended thesis is provided.

1.1. Overview

In this digital period, web has turned into an imperative wellspring of correspondence in very nearly every calling. The Internet is exceptionally efficient and cheap method of communication in every important field of life. With the expanded utilization of system engineering, its security has gotten to be extremely basic issue as the machines in distinctive connection holds very secure data and sensitive information. Be that as it may, the Internet Protocol (IP) on which the entire web is based is extremely insecure and vulnerable to viruses and hackers. Information Confidentiality and security is one of the significant issues for practically all connections particularly for Sensitive fields like Military, Avionics and Nuclear Power Centres. Commonplace numerous organizations confront new and unidentified sort of security dangers from different sorts of system gate crashers and programmers. System security has turned into one of the greatest tests for profoundly touchy connections.

1.2. Network Intrusion

A Network Intrusion is a suspicious and sudden deviation from the ordinary behaviour of the system. The Intrusion undermines the confidentiality, integrity and security of a network system. An Intrusion could be portrayed as "Any set of actions that attempt to compromise the integrity, confidentiality or availability of information resources" [1]. Network Intrusion joins unique kind of network attacks, data lost, worms, forbidden usage of methodologies and unpredictability in the standard behaviour of network traffic. On the other hand, the four critical sorts of network attacks are;

- **Remote to User assaults (R2l):** This is a strike in which the benefits of a client are misused on a remote PC. In this kind of attack, the ambusher does not have a record on a specific machine however he tries to establish unapproved access from a remote machine like speculating secret word [2]. Its cases are *xlock, guess_password, phf, sendmail, xsnoop* and so forth [6].
- **Denial of Service (Dos):** In this assault the memory assets are not permitted to react to demands to the system and web made by the clients in light of the fact that the assets are made excessively occupied by the gate-crasher to react to a true blue or quality solicitation. Hence the clients are not permitted to utilize the administration of their frameworks in this ambush. Its samples are apache, smurf, Neptune, back, mailbomb, udpstorm and so on [6].
- **User to Root Attacks (U to R):** In this strike the client endeavours to addition the benefits of directing client by means of a neighbourhood client account. It implies the assaulter as of recently has a nearby client account either lawfully or by utilizing unlawful methods, and afterward he tries to addition managerial access to the machine [4]. One of its cases is cradle flood in any structure [2]. The examples are Perl and xterm[6].
- **Probing:** In this ambush first the client distinguishes the shortcomings of the framework with the goal that they could be utilized to annihilate the framework later. It is carried out by picking up data about the host and system of machines. Its intention is to endeavour the security asset. Their cases are paragon of piety, *portsweep, mscan, nmap* and so forth [6].

1.3. Network Intrusion Detection

A Network Intrusion Detection System investigates the approaching network traffic and distinguishes the suspicious action on the system. Network Intrusion Detection is the most critical and most broadly utilized system security strategy used to recognize the vulnerable attacks by observing the system behaviour and afterward taking fundamental movements against them. Network Intrusion Detection systems are divided into host based and network based. This grouping relies on the information sources that are utilized [1]. Host Based system utilizes the histories and information maintained by the working system. By utilizing these histories, the system can screen things like framework logs; clients record records and document frameworks [5]. Network based Intrusion identification systems utilize network traffic data as its information source and screens the system activity [1]. So as to focus the behaviour of the system activity, the Network Intrusion Detection System must know the principles for normal action on the Internet so later they might be contrasted with the suspicious behaviour to recognize an attack. The Network Intrusion Detection System might be arranged into two classes depend upon the Intrusion discovery strategy; these classifications are given underneath,

- **Misuse Detection:** In misuse identification framework, the data acquired from the system is contrasted with a database of attack signatures. A signature is a set of rules that define a network attack. Misuse detection techniques are mostly used for commercial purpose or in industries having the network systems.
- **Anomaly Detection:** In aberrance identification framework, the ordinary conduct of web is characterized and afterward the system is screened by contrasting the activity and the characterized typical movement and conduct. Subsequently it recognizes the conduct that goes astray from the ordinary conduct and that conduct or connection is stamped as an ambush. Anomaly identification procedures are utilized for examination reason and scholarly territories for actualizing the network Intrusion detection systems because of its hypothetical capacity for tending to various types of attack [2].

1.4. Problem Statement

Throughout the past number of years, machine learning procedures have gained impressive consideration around the intrusion detection researchers to address the shortcomings of knowledge base detection systems. This has prompted the requisition of different supervised and unsupervised learning with the end goal of intrusion detection. Although there has been lot of research done in this field still there exist problems in identifying intrusions in the incoming traffic of network. These can be in the form of lack of accuracy of detection rate or it can behave differently for different types of attacks which lead to work well for one type of data but not for other hence, result in unbalanced detection rates and high false positives. Apart from that, data set have redundant input attributes as well as examples in the training data. The other delinquent part in this research area is to detect invasion attacks in high speed real time network because; the high-speed networks require intrusion detections systems to deal with large sizes of network data in a very short span of time.

1.5. Objectives

In this thesis, implementation of machine learning techniques to intrusion detection has been presented. Two learning algorithms, Naïve Bayes and Self Organizing Map have been chosen for the assignment of detecting intrusions and their relative exhibitions with different combination of data pre-processing techniques. There is specifically extracted information situated is KDD data set with the end goal of experiments with intrusion detection problems. KDD information set hold 42 connection records. As a rule intrusion detection framework utilizes all the connection records accessible in the information with the end goal of the intrusion detection. Utilizing all the connection records of the information set increases learning time of the calculation implies late identification of the intrusion by IDS. Here, we are proposing the characteristic decrease of the information set utilizing data pick up. After reduction of the feature, learning time of the algorithm decreased drastically without compromising the accuracy of the IDS which is desirable. We finally state the usefulness of machine learning to the field of computer security and also comment on the security of machine learning itself.

1.6. Scope of Thesis

The essential thought in this theory is to utilize two various machines learning procedures to break down the system behaviour and recognize the suspicious and attacked connections. The machine learning strategies can be supervised and unsupervised learnings. For supervised machine learning technique, Naïve Bayes Algorithm is prepared on the dataset and for unsupervised learning; Self Organizing Map is prepared on the dataset. By utilizing two separate strategies, the effectiveness and precision of every method might be distinguished in Intrusion Detection System.

The dataset picked is KDD Cup 1999 dataset as it is particularly designed for the Intrusion detection issues. This dataset was concentrated from the 1998 DARPA Intrusion Detection Evaluation Program which was ready and managed by MIT Lincoln Labs. This dataset holds numerous connection records which have symbolic and numeric qualities. Another form of the dataset NSL-KDD is utilized for the usage of Intrusion identification frameworks. This dataset has made a considerable measure of change in the KDD Cup 1999 dataset [7]. This dataset holds symbolic and numeric characteristics. The numeric characteristics additionally hold large range of values. The symbolic characteristics additionally need to be changed over to numeric with a specific end goal to be utilized by machine learning procedures. So the dataset is pre-processed to scale the large range of values to smaller ranges and symbolic characteristics are changed over to numeric characteristics utilizing symbolic conversion systems. For the pre-processing of symbolic characteristics of the dataset in this work, two separate sorts of symbolic conversion techniques are utilized which are Conditional Probability and Indicator Variables Conversion techniques. After training the effect of symbolic conversion along with its particular machine learning algorithm are compared on the test dataset.

1.7. Organization of the Thesis

The remainder of the thesis is organized into 7 Chapters further as follow;

Chapter 2 introduces the general concepts of Intrusion Detection Systems.

Chapter 3 highlights the Background and Related work

Chapter 4 explains the Supervised and Unsupervised Machine Learning Approaches

Chapter 5 describes Dataset KDD CUP99

Chapter 6 describes our implementations of the two algorithms we are using: Naïve Bayes and Self-Organizing Maps.

Chapter 7 presents the results of our experimentation and discusses the different findings.

Chapter 8 concludes the thesis by summarizing the outcomes and suggesting some ideas for further work.

CHAPTER 2
INTRUSION DETECTION SYSTEMS

2.1. Introduction

This chapter anticipates giving a speculative framework and prologue to Intrusion detection systems. First and foremost, the significance of computer security is discussed. Following this, different possible threats to security are identified. A depiction of IDS is then provided along with a detailed description of its components. The IDS is further discussed along with its different methodologies and types along with its application.

2.2. Computer Security and its Role

One wide importance of an ensured machine framework is given by Garfinkel and Spafford [14] as one that could be depended on to go about as it is depended upon to. The dependence on the typical behaviour being the same as indicated behaviour is alluded to as trust in the security of the machine structure. The level of trust exhibits the assurance in the ordinary behaviour of the PC system. The typical conduct is formalized into the security methodology of the workstation schema and administers the destinations that the system must meet. This methodology may fuse reason necessities accepting that they are fundamental for the feasible working of the PC framework.

A narrower importance of machine security is disadvantaged upon the affirmation of privacy, dependability, and openness in a PC structure [15]. Characterized obliges that information be responsive simply to those appointed for it, reliability obliges that information remain unaltered by disaster's or toxic attempts, and openness suggests that the machine system remains working without degradation of access and offers advantages for affirmed customers when they oblige it. By this definition, a dishonest PC system is unsecure if availability is a part of its security requirements.

An ensured workstation structure guarantees its data and possessions from unapproved get to, changing, and difference of use. Grouped data may be crucial to the business success or survival of an association, data respectability may be principal to a mending focus that regulates remedial histories of patients and usage of it to settle on life fundamental decisions and data openness may be significant for steady development control.

There is a close relationship between the utilitarian exactness of a PC framework and its security. Useful exactness construes that a PC framework achieves its determinations. In case the convenience specific fuses security technique necessities, then useful rightness proposes security of the PC system. Of course, the opposite is not faultless, i.e., utilitarian slip may not realize violations of the security approach, especially as it relates to mystery, uprightness, and openness. For example, a working system organization call may not set up all considerable disputes to it adequately, yet it may not be possible to misuse the security approach by misusing this. As an interchange outline, contemplate a visual (WYSIWYG) word converting program that fails to highlight customer determinations on the showcase. The undertaking is likely not practically right; however this behaviour may not result in a violation of the structure security system.

2.2.1. Threats to Security

As a social we are becoming logically dependent on the quick access and provision of information. As this input has extended, more information may be undoubtedly filed on workstations. The extended use of workstations has made brisk connection of data from unique sources possible. Correspondence of information from notable sources has allowed additional information to be accumulated that may be trying to procure particularly. The extension of sensible PCs and of machine frameworks has exacerbated the issue of unapproved access and messing around with data. Stretched connectivity not simply offers access to greater and changed stakes of data more quickly than at any possible time, it moreover outfits a right to further enchant route to the data from essentially wherever on the framework [16].

In many cases, for instance, the Internet worm strike [17], framework gate crashers have easily overwhelmed the password authentication proposed to secure schemas.

With an extended perception of schemas capacity, intruders are skilled at choosing inadequacies in structures and harming them to get such stretched benefits that they can do anything on the framework. Interlopers also use illustrations of interference that are difficult to look after and recognize. They generally use a couple of levels of indirection before breaking into target systems and occasionally revel in sudden impacts of suspicious or odd development. They also cover their tracks with the objective that their development on the entered framework is not adequately revealed.

Dangers, for instance, contaminations [18] and worms [19] do not oblige human supervision and are fit for reproducing and making an excursion to joined workstation frameworks. Unleashed at one PC, when they are uncovered it may be troublesome to take after their beginning stage or the level of ruining. By then there may be dangers from Trojan stallions which do not impersonate however are altered to unleash ruinous development on a precondition amassed into the framework [20].

2.2.2. Identifying Threats

Most workstation systems give a right to increase passageway control part as their first line of boundary [21]. However, this only limits whether access to an object in the system is permitted but does not model or restrict what a subject may do with the object itself if it has the access to manipulate it [22], Access control hence does not model and cannot hinder unapproved information travel through the structure since such stream can happen with dispatched enters to the articles. Additionally, in systems where access controls are no compulsory, the commitment of guaranteeing data rests on the completion customer. This consistently obliges that customers appreciate the protection instruments offered by the frameworks and how to achieve the fancied security using these parts.

Information stream could be directed to update security by applying models, for instance, the Bell and Lapadula model [23] to outfit riddle, or the Biba model [24] to give trustworthiness. Then again, security takes a swing at the risk of solace. Both models are traditionalist and cut off points read and create operations to certification that confidentiality and reliability of data in the structure cannot be exchanged off. Expecting that both models are commonly used, the resulting model simply permits increases doorway to inquiries at the same security gathering level as the subject. Subsequently, a completely secure schema may not be outstandingly profitable.

Access controls and security models are not steady against insider dangers or deal of the confirmation module. On the off chance that a watchword is frail and is dealt, access control measures cannot expect the hardship or degradation of information that the exchanged off customer was authorized to get access to. When in doubt, static schedules for ensuring security arrive in a framework may essentially be needed, or make the system unnecessarily restrictive to its customers. For example, static frameworks will doubtlessly be unable to prevent violation of security approach that happens due to looking of data reports; and obliged access controls [25] that simply permit customers access to data for which they have suitable flexibility and make the schema cumbersome to use. An element strategy is accordingly needed to run across and perhaps turn away breaks in security.

The tests in building intricate, without bug writing computer programs are farfetched to be determined inside a short compass of time. Accuses in system writing computer programs are customarily indicated as security weaknesses. Other than, programming life process lengths of time are continually incessantly contracted in view of stretched business power. This as often as possible achieves poor arrangements or lacking testing, further disturbing the issue.

Machine schemas are subject to remain unsecure for very much a while to come. We must have measures set up to get security breaks, i.e., distinguish intruders and intrusions. Intrusion ID systems fill this part and typically structure the last line of gatekeeper as a rule protection arrangement of a PC system. They are useful in uncovering extraordinary breaks of security, and in directing attempts to break security, which outfits vital information for helpful countermeasures. Thus, intrusion detection systems are useful even when strong preventive steps taken to protect computer systems place a high degree of confidence in their security. Furthermore, preventive steps such as repairs of system software faults may not always be preferable to detection of their exploitation from a practical cost-benefit consideration. Settling bugs may not be possible without the item source and basic authority, and tremendous scale plan of patches may require more cumbersome establishment methods than overhauling the interference area database, especially when writing computer programs is changed for neighbourhood use at unique objectives. By virtue of sweeping, complex undertakings, for instance, send letters; it may not be possible to "change" all its possible defects really when its source code is available. Screening non particular frameworks for misusing vulnerabilities could be greatly convenient in such cases.

2.3 Diagrammatic Representation of PC Frameworks with and without IDS:

Having analysed the security threats to unprotected PC frameworks, the following diagrams represent two different scenarios in order to better understand the application of IDS.

2.3.1 Scenario I

Unsecured PC Framework representation

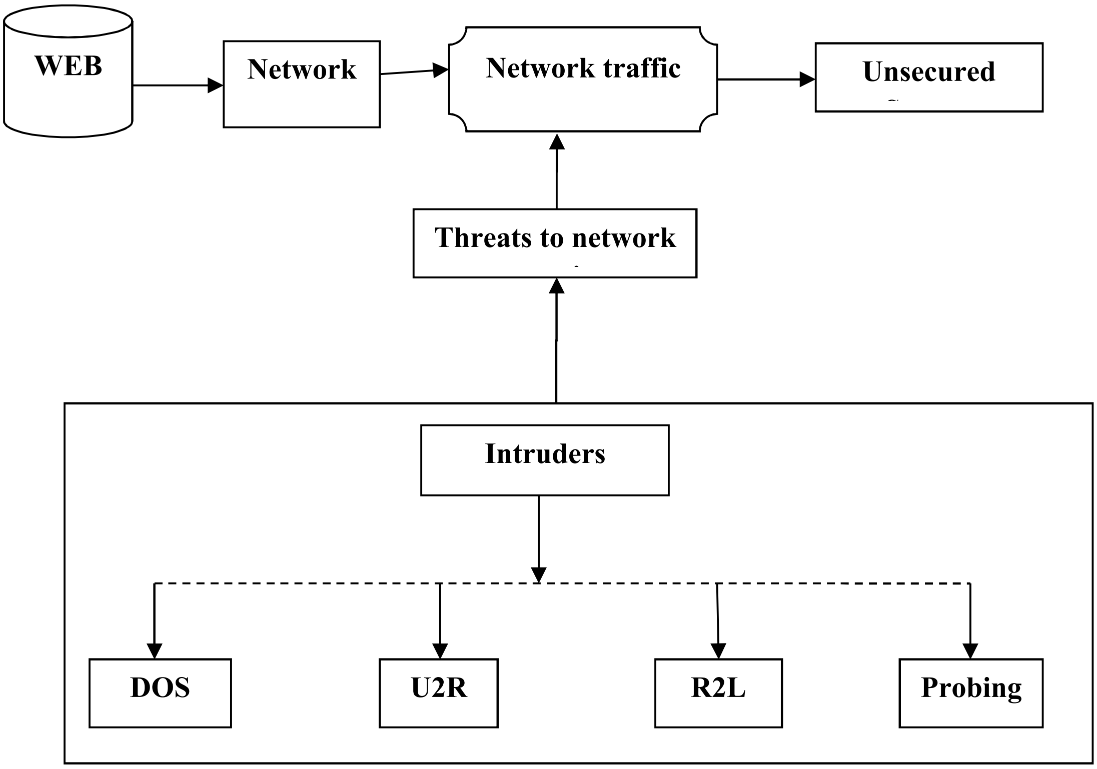


Figure 2.1 - **Unsecured PC Framework representation**

2.3.2 Scenario II

Secured PC Framework representation with the help of IDS

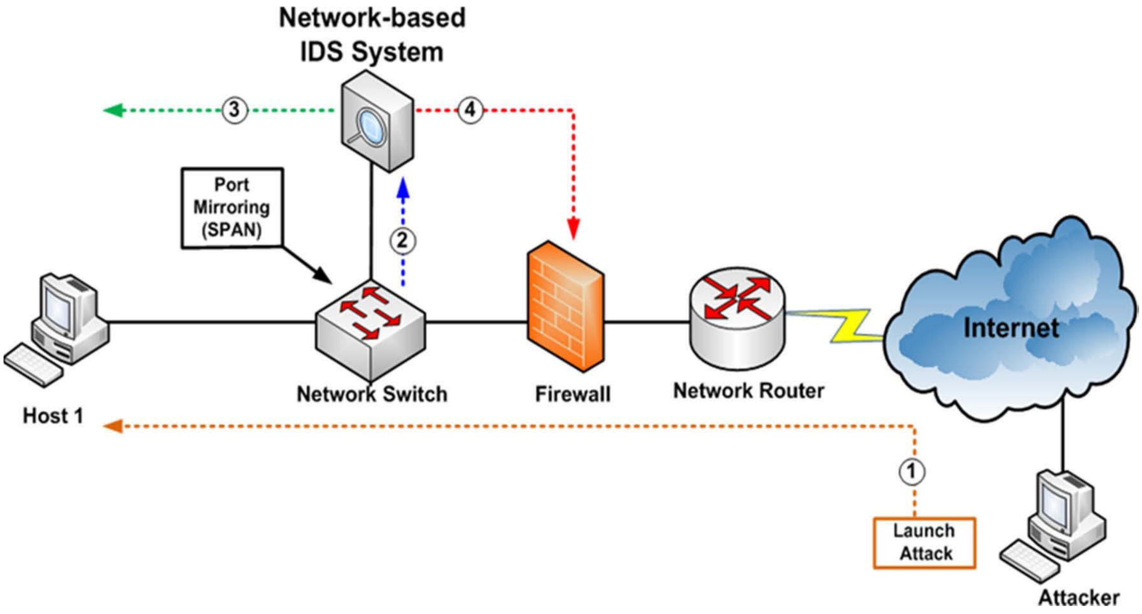


Figure 2.2 - **Secured System with IDS**

2.4 What is Intrusion Detection?

This section provides a brief literature review on the IDS, its various components, types followed by its applications:

2.4.1 Introduction

An intrusion is portrayed by Heady et al. [26] as

"any set of developments that try to compromise secrecy, or availability of a resource".

A former study finished by Anderson [28] uses the outflow "hazard" in this same sense and describes it to be:

"The potential likelihood of an arranged unapproved try to

- Access information,
- Manipulate information, or
- Render a system conflicting or unusable.

Interference is a violation of the security methodology of the system. The definitions above are general enough to fuse all the dangers said in the past portion. Any importance of intrusion is, of need, unverifiable, as security game plan requirements do not reliably make as a by and large described set of developments. Because of the fact that approach describes the goals that must be satisfied in a system, finding bursts of technique obliges data of steps or exercises that may achieve its violation.

2.4.2. Network Intrusion Detection Systems

A Network Intrusion Detection system is an intrusion detection structure that tries to recognize noxious movement, for instance, repudiation of organization trap, port compasses or attempts to break into workstations by supervising network traffic.

A Network Intrusion Detection system scrutinizes all approaching packs and tries to uncover suspicious illustrations presumed to be checks or fundamentals. These standards are picked by a framework chief while the course of action and sending of the Intrusion detection systems subordinate upon the security and framework plans of the companionship. Case in point, expecting that it is viewed that a particular TCP acquaintanceship requests companionship with innumerable, then it could be acknowledged that there is someone who is endeavouring to administer a port yield of all/the vast majority of the workstations of the framework.

A Network Intrusion Detection system is not obliged to surveying the approaching framework movement just. Illustrations and neighbourly interference can moreover be found from the heartfelt or close-by development as well. A couple of ambushes may moreover hail from inside the watched framework, as in trusted host strike [28].

2.4.3. Evolution of Intrusion Detection

A couple of people have studied the state of the art, including:

James Anderson, 1980

The origin of intrusion area began with James Anderson's specific report, Computer Security Threat Monitoring and Surveillance for the U.S. flying corps. The paper showed that survey records could be used to distinguish machine ill-use and to recognize danger requests, and it offered recommendations to improve inspecting of systems to recognize misuse. It is subordinate upon the feelings that an intruder's behaviour will be perceptibly not exactly the same as that of a genuine customer and that various unapproved developments will be observable.[27]

Dorothy Denning, 1985

This study introduces a model that may research survey trails from government frameworks and track customer activity. He named this system as Intrusion Detection Expert System (IDES), and it was the foundational investigation into IDS building.[30]

Dorothy Denning, 1986

This study presents the first interference revelation model, which has six guideline parts: subjects, articles, audit records, profiles, and variation from the norm records, and movement fundamentals. Subjects imply the initiators of development in an information framework; they are for the most part commonplace customers. Articles are the advantages administered by the information system, for instance, records, summons and contraptions. Survey records are those made by the information schema due to exercises performed or tried by subjects on articles. [30]

Dorothy Denning, 1987

This study presents profile of the behavioural examination. Behavioural examination hunt down deviations from the kind of direct that has been accurately base lined, for instance, connections in groups and in what is persistently sent over a framework. [31]

Haystack, 1988

This study presents a joined together aberrance detection/misuse discovery IDS that models distinctive clients and gatherings of clients. It allocates introductory profile to new clients, and upgrades the profiles once an example of genuine conduct is distinguished. Bundle utilized as an alternate measurable anomaly detection system.[32]

Lunt, T.F., 1988

This study displays a joined together aberrance identification/misuse revelation IDS that models notable customers and social events of customers. It assigns starting profile to new customers, and redesigns the profiles once an illustration of veritable behaviour is recognized. Group used substitute measurable characteristic detection estimation.[33]

Todd Heberlein, 1989

An IDS framework called Network System Monitor (NSM) was displayed by Todd. NSM was remarkable in connection to IDES and DIDS (Distributed Intrusion detection systems) in that it may explore framework movement instead of system logs. NSM, nearby the now monetarily available Stalker IDS, served to make new care and excitement to IDS research for the business and open portions.[34]

Marcus J. Ranum, 1990

This study presents business IDS called Network Flight Recorder (NFR). Christopher Klaus and Thomas E. Noonan built Internet Security Systems (ISS) and released a framework based interference recognizable proof structure called Real secure. [35]

Teng Chen, And Lu, 1990

This study exhibits the usage of a period based inductive machine (TIM) to get a customer's behaviour outline. As a generally convenient gadget, TIM discovers transient progressive illustrations in an aggregating of events. The progressive samples address significantly dismal activities and are depended upon to give predication. The short lived samples, which are spoken to as precepts, are made and modified from the data using a sensible construing, called inductive generalization. The moment that associated with intrusion finding, the standards portray the behaviour samples of either a customer or a social occasion of customers needy upon past audit history.[36]

Mukherjee, B. Heberlein, L., and Levitt, K, 1994

This study examined a couple of host-based and framework based IDSs, and distinguished the properties of the analysing frameworks are recognized. The host-based systems use the host meeting expectations system's audit trails as the essential wellspring of information to find intrusive movement, while the dominant part of the framework based IDSs manufacture their area segment by administering network traffic, and some use host survey trails likewise. A diagram of a verifiable conflict disclosure computation used in normal IDS is similarly included [37].

Anderson D, Lunt TF, Javitz H, Tamaru A, Valdes, 1995

This report indicates the findings of System Research Institute's (Sri's) examination of the Network Intrusion- Detection Expert System (NIDES). The true examination heeled to place unapproved demand execution indigent upon schema level audit trails.[38]

Sundaram, A., 1996.

A graph of interference disclosure thoughts and experimental grouping was given. Next, it displays and looks at a couple of business and open region IDS's available. This study furthermore depicts late headways in reliable interference finding: Distributed, measured system which fuses both conflict and misuse distinction. A gander at the new sort of virtuoso energized, safety measure instruments implied Delphic gadgets recognize the dangers and dangers in the precise early strike stages; was also discussed.[39]

Axelsson, S., 1998

This study depicts the thought of interference distinguishing proof in neighbourhood, connected with the Internet. The intrusion ID systems and vulnerabilities in interference distinction structures (IDS) and also intrusion revelation and response (IDR) frameworks were explored. A capability is made between host-based and framework based interference schemas. The framework security standards, general connectivity and comparability necessities of IDS are moreover inspected. [40]

Lane and Brodley, 1998

This study exhibits the associated sample based learning (IBL) to take in components' (e.g., customers) common behaviour from momentary game plan data. IBL identifies with a thought of energy with a set of cases that depict the thought. The set of events is known as the case dictionary. An alternate event is assembled as expressed by its connection with set away events. IBL obliges a considered "division" between the cases so that the closeness of assorted events could be measured and used to mastermind the illustrations.[41]

Lee W. and Stolfo S. and Mok K, 1999

In this document, a data burrowing mapping for adaptively manufacturing Intrusion Detection (ID) models was depicted. The central thought was to utilize assessing tasks to focus an extensive set of aspects that portray every one framework.[42]

Allen, J., Christie, A., and Stoner, 2000

This manuscript indicates an exhaustive review of oddity area structures and hybrid intrusion distinction schemas. The late inventive examples in aberrance distinction and distinguish open issues and tests around there were similarly analysed.[43]

Debar, H., Dacier. M. And Wespi, 2000

This tabloid discussed an experimental arrangement of intrusion distinction frameworks that highlight the diverse parts of this zone. This investigative order differentiates gatherings of intrusion recognizable proof schemas as expressed by their properties.[44]

Alessandri, D, 2001

This manuscript displays the mix of the experimental arrangement introduced and the thought of development amplification outfits us with a versatile and feasible instrument to portray intrusion revelation structure. This investigative characterization has been transformed to depict the convenient perspectives i.e., the limits of interference distinction schema such that one can in a next step survey intrusion recognizable proof structure for their probability to find strikes, transform false positives etc. [45]

Axelsson, S, 2002

An exploratory classification of interference distinction structures was presented. The investigative arrangement involves a gathering first of the distinction standard, and second of certain operational parts of the interference area structure. The frameworks are moreover collected as expressed by the expanding test of the issue they attempt to address. These requests are used predicatively, showing towards different zones of future research in the field of interference finding. [46]

Bai, Y., And Kobayashi, H, 2003

In this tabloid a review on genuine tests to ID building was shown. It is incorporated with a couple of guideline parts of ID advancement. Breaks down on interference ID systems and data gathering frameworks are accentuated. Some original progressions in ID Systems, for instance, both data mining based ID frameworks and data consolidation based ID schemas, were also analysed. Current ID designing faces viable tests.[47]

Yu Chen, Yu-Kwong Kwok, and Kai Hwang, 2005

This manuscript shows an alternate sign changing approach to recognize and distinguish the ambushes (by reviewing the repeat zone qualities of approaching development streams to a server). This technique is effective in that its area time is short of what several seconds. Furthermore, this system includes direct execution, making it deployable in real framework circumstances.[48]

Phillip Brooke, 2006

This tabloid shows an alternate IDS structure for compact adhoc framework (MANET) circumstances based upon the thought of a buddy in a little world sensation. The two-level IDS structure has been proposed to overcome longer ID instruments and revelation encountering the potential for intimidation aggressors and false attributions with the aid of buddy centre points. It is hypothesized that with the presentation of buddy centre points, the impacts of the IDS issues could be minimized. It is noted that the proposed pattern may not have the ability to chip away at a contrasting MANET circumstances.[49]

Jeyanthi Hall, 2007

This tabloid shows a discontinuity based interference distinction approach, which combines radio repeat fingerprinting (RFF) and Hotelling's T 2, a multivariate genuine procedure control method, for distinguishing this pitfall. RFF is a framework used to uncommonly recognize a transceiver subordinate upon the transceiver print (set of qualities) of the marker it produces. The approach is to partner a transceiver profile of a remote device with its analysing MAC address. In this way, in spite of the way that the MAC area can regardless be criticized, the transceiver prints from the illegitimate contraption may not match the profile of the true blue device. Additionally, the triumph rate of remote IDS could be further improved, by inspecting various consecutively asked for transceiver prints, going before rendering a decision. [50]

Eduardo Mosqueira - Rey et al, 2007

This manuscript depicted the arrangement of misuse area driver which is one of the assorted agents in a multivalent-based interference recognizable proof system. Using a bundle sniffer the agent examines the packs in the framework acquaintanceships and makes a data model subordinate upon the information gained.[51]

Magnus Almgren, Ulf Lindqvist, and Erland Jonsson, 2008

This study inspected the strategy to use the alerts from may audit sources to upgrade the effectiveness of the interference area system (IDS). A theoretical model was arranged commonly for the reason about the alerts from the dissimilar sensors through concentrating on the web server ambushes. It also gives a better understanding of possible ambushes against their structures for the security experts.[52]

Naeimeh Laleh and Mohammad Abdollahi Azgomi, 2009

This study discussed the deception that is creating astonishingly with the advancement of state-of-the-art building and the broad superhighways of correspondence which realizes the adversity of billions of dollars all as far and wide as could be allowed consistently. This methodology tends to propose an alternate investigative order and complete review for the various sorts of trickery and data mining frameworks of tricking distinguishing proof. [53]

Yurong Xu, James Ford, and Fillia Makedon, 2010

This manuscript shows dispersed wormhole ID estimation called Wormhole Geographic Distributed Detection (WGDD) that is needy after finding framework issue launched by the vicinity of a wormhole. Since wormhole strike are uninvolved, this estimation uses a hop recognizing system a test technique to distinguish wormhole ambushes, then duplicates neighbourhood maps in every centre point. After that, it uses a trademark called "estimation" to distinguish aberrances by wormholes. The guideline purpose of enthusiasm of using an appropriated wormhole area computation is that it gives the assessed zone of a wormhole, which may be of administration information for further watch parts.[54]

2.4.3 IDS Components

The Common Intrusion Detection Framework (CIDF) is a commonly adopted system that portrays the working of an Intrusion Detection Systems (IDSs). The CIDF divides IDS into four components, as indicated in Figure 2.1, which are [28]:

- Sensors
- Analysers
- Incident space
- Response units

These fundamental segments of IDS will be depicted in additional detail underneath.

2.4.3.1 Sensors

The part of the sensors is to acquire cautions from the bigger computational environment outside the intrusion detection system, and give them in the CIDF to whatever is left of the framework [28]. These occasion generators are frequently alluded to as occasion generators.

Most current intrusion detection systems utilize different intrusion sensors to maximise their dependability. It is essential to circuit the distinctive yields of these sensors in a powerful and clever way with a specific end goal to furnish the security chairman with an "in general security see" that can serve as an associate to assess the dependability in the multi-sensor IDS [29].

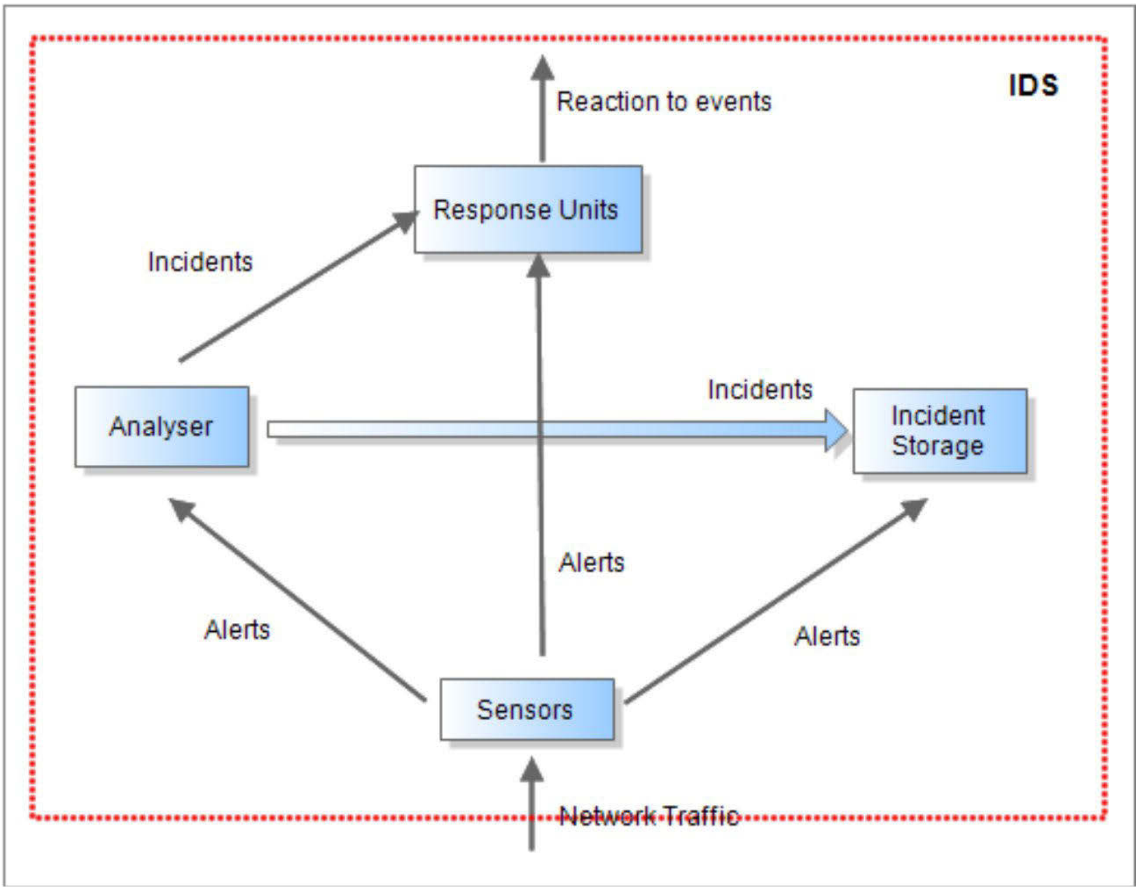


Figure 2.3 - **Common segments of the Intrusion Detection Framework.**

By general discussion, there are two notable sorts of sensors; network based sensors and host-based sensors.

2.4.3.2 Analysers

Analysers utilize the information of the sensors, break down the data accumulated by these sensors, and return an amalgamation or outline of the information [28]. Today, fake intelligence has turned into a crucial instrument in the analysers of intrusion detection systems [29]. There are two sorts of analysers; pattern based analysers and anomaly based analysers.

2.4.3.3 Incident Space

The incident space stores all cautions, supporting the examination process. Throughout the investigation methodology occasions of alarms saved in the incident space might be utilized as a part of request to characterize a caution. Databases are not anticipated that will change or methodology the alarms in any method [28].

2.4.3.4 Response Units

Response units (either teams of humans or automated processes) carry out prescriptions controlled by the analyser that instructs them to act [28]. The reaction unit might for instance send an email to a framework director to advise him/her of suspicious cautions. An alternate conceivable reaction could be to adjust firewall leads keeping in mind the end goal to forestall further intrusions.

2.4.3.5 Goals and Capacities of IDS

Numerous studies have demonstrated that most machine security incidents are brought about by insiders, i.e. individuals who might not be obstructed by a firewall. Since insiders oblige access with huge benefits to do their everyday occupations, this outcomes in the necessity for additional efforts to establish safety inside the organisation.

IDSs may supplement other preventive controls (e.g. firewalls) as the following line of defence inside the organisation [55]. An Intrusion Detection System is a gadget that is set inside a secured system to screen what happens inside the system. This is schematically indicated in Figure 2.2. An Intrusion Detection System offers the chance to recognize an aggressor that has the ability to pass through the switch and pass through the firewall. The detection can happen at the start of the strike, throughout the ambush, or after it has happened. IDSs enact a caution, which can make protective move [56].

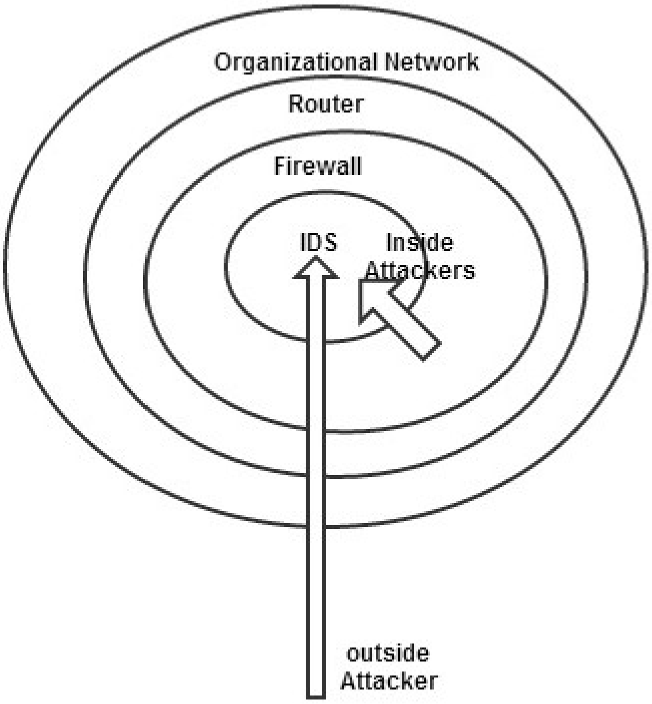


Figure 2.4 - **Arrangement of an Intrusion Detection System inside an organisational system [56].**

The objective of intrusion detection systems is to faultlessly distinguish abnormal system conduct or misuse of assets (i.e. occurrences), deal with correct assaults from false alerts, and advise system chairmen of the movement. Numerous organisations now utilize Intrusion detection systems to help them focus if their frameworks have been traded off.

Given the objective of IDS, the capacities of IDS might be [55]:

- Monitoring clients and network traffic
- Auditing framework arrangement for vulnerabilities and misconfigurations
- Assessing the uprightness of basic framework and information records
- Recognising known assault designs in framework movement
- Identifying irregular through statistical investigation
- Managing review trails and highlighting client violation of arrangement or ordinary movement
- Correcting framework design slips

- Installing and working traps to record data about gate crashers

IDS may epitomize one or a greater amount of these capacities, contingent upon the kind of IDS. The above mentioned capacities help the organisation's security groups, for example, Computer Emergency Response Teams (Certs), otherwise called Computer Security Incident Response Team (CSIRT).

2.4.4 IDS Methodologies

Both for the information gathering (or occasion generators) and in addition the investigation of the cautions, there are two methodologies. For the accumulation of information, there is a network based IDS methodology, and a host- based IDS approach. For caution dissection, IDSs may be either signature-based or heuristic. Remembering the deciding objective to build an enhanced understanding into current IDSs, all these dissimilar methodologies will be said rapidly. In the wake of having discussed the previously stated philosophies, the inside will be on one specific kind of utilization; honey pots.

2.4.4.1 Network Based IDS

A network based IDS is a stand-alone mechanism appended to the system to screen movement all around that system [55]. Intrusion detection is recognized to be network based when the framework is utilized to gather and investigate system bundles. Framework groups are regularly "snuffle" off the framework, and are preferably construed straight from the yield of switches and switches [57].

2.4.4.2 Host-Based IDS

A host-based IDS runs under a customer or host workstation to secure that particular host [55]. Intrusion detection is host-based when the framework is utilized to examine information that starts on PCs (hosts, for example, requisition and working framework occasion logs. This is instead of network based intrusion area, is used to process data that begins on the framework. Host-based intrusion area is particularly constraining at placing insider misuse because of the target data source is region to the checked customer [57].

2.4.5 Types of IDS

There are two sorts of Intrusion detection systems; pattern based and anomaly based frameworks [82]. The contrast between these two frameworks lies in the kind of discovery utilized throughout the investigation stage specified in Figure 2.2.

2.4.5.1 Pattern Based Systems

Pattern based, also referred to as signature-based, IDS perform straightforward pattern matching and report circumstances that match a pattern comparing to a known attack class. Anomaly based IDSs, otherwise called heuristic IDSs, construct a model of worthy conduct and banner (i.e. name) special cases to that model; for what is to come, the overseer can check a hailed conduct as worthy so the anomaly based IDS will now treat that formerly unclassified conduct as adequate [55][32]. Most business IDSs utilise this methodology [59].

The fundamental preference of pattern based systems is that it has a generally low rate of false cautions [82], which implies that the IDS have a moderately high accuracy. This high exactness is brought about by the way that a pattern based framework is expressly modified to discover certain known sorts of strike [82].

The fundamental detriment of this sort of frameworks, in any case, is that the identification rate of strike is moderately low [82]. Inferred from that, one could accept an easier review for new sorts of intrusions. This is the after effect of the accompanying two issues of pattern based frameworks.

The main issue with signature based detection systems are the signatures themselves. An ambusher will attempt to alter a fundamental strike in such a path, to the point that it will not match the known signatures of that assault. The ambusher may embed contorted parcels that the IDS will see, to purposefully cause an example confuse; the convention handler stack will then toss the bundles in light of the malformation. Each of these varieties could be caught by IDS, yet more diverse signatures oblige extra work for the IDS, which decrease execution [55][32].

Moreover, pattern based IDSs cannot uncover an alternate trap for which a signature is not yet introduced in the database. Ideally, patterns should match every sample of a snare; match unpretentious mixtures of the light up, not portable fire stick development that is not some bit of a strike. Be that as it may, this goal is trying to complete in present IDSs [32].

2.4.5.2 Anomaly Based Systems

Since signatures are restricted to particular, known assault designs, supposed heuristic intrusion detection gets suitable. As opposed to searching for matches, heuristic intrusion detection searches for conduct that is suspicious [55][32]. Inconsistency based frameworks endeavour to characterise typical operation, and attempt to catch any deviation from ordinary conduct [82]. This type of intrusion detection is otherwise called Anomaly Intrusion Detection. The standard of this kind of intrusion identification is that the true action is thought about against a known suspicious zone of cautions [55].

The preference of an anomaly based intrusion detection system is a generally high detection rate for new sorts of intrusions, i.e. a higher review. The fundamental disservice of anomaly based intrusion detection, then again, is a higher rate of false cautions too, which implies an easier exactness [82]. An alternate detriment is that this sort of IDSs has a tendency to be computationally costly since some measurements are frequently looked after that need to be overhauled against each framework movement and, because of deficient information, they might continuously be prepared erroneously to recognise a meddling conduct as ordinary because of lacking information [59].

2.4.6 Applications of Network Intrusion Detection Systems

As discussed in section 2.2, accepted security is not satisfactory for contemporary systems. With expanded use and more instances of intrusion occurring now- a-days than at any other time in the recent past, we need something to improve the security. This is the place a network intrusion detection system comes into the picture.

The accompanying are the essential needs that a network intrusion detection system can satisfy [60].

a) Backing up firewalls: In numerous cases, interlopers attempt to and enter firewalls to addition unapproved access to corporate systems. This is carried out by assaulting the firewall itself and separating it by tweaking its leads and signatures. Thus, the network intrusion detection system can diminish the danger of such strike by briefly moving down firewalls. The network intrusion discovery arrangement of this sort channels bundles dependent upon their IP parcel header. This empowers the system overseer to send Network Intrusion Detection systems with usefulness similar to that of exceptionally developed firewalls. Further, this sort of network intrusion detection system can likewise be utilized while the general firewall is down for support or when the firewall programming is constantly overhauled or for any viable reason.

b) Controlling index access: Generally capacities of regulating document access are carried out to concentrated frameworks, for example, Secret Net, which are expected particularly for ensuring system data from unapproved access. Nonetheless, assurance of some basically vital records, for example, database indexes and secret word documents cannot be carried out by such frameworks. Additionally, such frameworks are essentially produced for the Windows and Netware stages. So such frameworks fall flat in UNIX situations which are utilized for system provisions within numerous connections. So in such sorts of cases a network intrusion detection system acts the hero of system executives. For the most part have based network intrusion identification frameworks are utilized within such cases which are built both with respect to log-record investigation (Real Secure Server Sensor) and IDSs breaking down framework calls (Cisco IDS Host Server).

c) Controlling the Administrative Activities: Network Intrusion detection systems can go about as an extra control device which can check unapproved arrangement changes that have been allowed regulatory benefits.

d) Protection against Viruses: There has been a disturbing build in the amount of infections and worms that now attack the web and influence various Pcs ordinary. Worm plagues like the Red Code, Blue Code, Nimada and so on has exhibited the threat of disparaging the dangers of such vindictive projects.

Network Intrusion detection systems could be supportive in these cases also. In spite of the fact that they cannot supplant the universal antiviral programming yet they can go about as an extra boundary for infections, worms and Trojan stallions.

e) Detecting Obscure Apparatuses: A network intrusion detection framework can help in distinguishing the detection of unknown/external has inside the secured system sections. It can likewise locate expanded movement and uncommon sort of exercises from particular workstations which were not included in such sort of exercises some time recently. Such exercises might be a clue to malignant exercises from the hosts and the system director must be educated about this.

f) Analysing the Yield of Firewall Settings: Firewalls are fundamental for securing the corporate system from unwanted system exercises. Be that as it may a firewall can work attractively just when it is designed accurately. Inaccurate setup and wasteful testing of a firewall can wreck destruction on the system. Introducing a network intrusion detection system first and then the firewall permits one to test the productivity of the firewall by analysing the amount of strike prior and then afterward the firewall. Notwithstanding this, it can likewise go about as reinforcement for the firewall.

g) Analysing the Data Streams: Situations where the specialized masters have no dependable data on the conventions utilized within the secured system fragments. A network intrusion detection system can control all the conventions and administrations utilized as a part of the corporate system, and in addition the recurrence of their utilization. This empowers to make a plan of data stream and the system map [61]. This production of data streams and the system guide is a crucial necessity for effectively making a data security base.

h) Analysing Information from the System Equipment: Log documents from switches and other system gear can serve as an extra wellspring of data on the different strike that an information system might be inclined to. Nonetheless, most connections do not investigate these gathered data on the grounds that it is a period overhead for the connection and the devices accessible for such breakdowns, (for example, netforensics) are noticeably unreasonable.

A network intrusion detection framework might be arranged to do this work. The undertaking of gathering such log-document data and examining logged security occasions could be appointed to the intrusion detection framework, which hence, serves as a Syslog server. It can concentrate such errands of gathering log-index data and identify assaults and misuse of the system. It likewise averts unapproved changes of the occasions logged. Also, the occasions logged are quickly sent to an alternate server with the goal that the gate crasher cannot uproot any follow in the wake of finishing her operation.

i) Collecting Verification and Taking care of Occurrences: Network Intrusion detection systems can, and might as well; be utilized for gathering confirmation of unapproved action in the system either by trusted hosts or outside culprits. This is proficient by the network intrusion identification framework with the accompanying functionalities-

I. Logging the occasions that happen throughout an ambush in a database or an outside index which could be dissected later.

II. Imitating requisitions which do not exist to mislead the interloper. This sort of usefulness is otherwise called the trickiness mode [60] of working of the network intrusion detection framework.

III. The network intrusion identification framework, notwithstanding logging occasions, additionally help in upgraded investigation of those log documents which were made either by the framework or the provision programming or database and web servers.

IV. The network intrusion detection system can likewise help in getting data of the gate crasher like his DNS, MAC, Netbios and IP address.

j) Performing Stock and Making a System Delineate: Outline demonstrates as its subject essential connections inside a system, for example, streets, tram lines, pipelines, or landing strip connections [61]. A network intrusion identification framework could be utilized to make a system map which might be utilized to assemble different data about system hosts and gate crashers. The data that could be gathered are [60].

I. The part of the host and its DNS and NETBIOS names.

II. Network administrations

III. Active administration headers

IV. Types and variants of operating systems and requisition programming.

V. Netbios offers

VI. User and administration accounts

VII. General parameters of the security strategy review arrangement, client and watchword approach et cetera.

k) Detecting Default Setups: Most system heads utilize the default system designs for improving their undertakings. In any case this additionally streamlines the errand of a gate crasher on the grounds that she knows the default system designs. This makes the system more vulnerable and open to fruitful strike. A network intrusion identification framework could be arranged to hunt the hosts where default arrangements have been utilized and can likewise prescribe restorative measures that might be taken.

2.5 Summary

From the above discourse, Intrusion detection is an essential segment of the security controls and mechanisms furnished in a framework. It normally structures the last line of protection against security dangers. These components are planned to identify ruptures of approach that cannot be effectively identified utilizing different systems. Intrusion detection is normally dependent upon one of two models: anomaly and the misuse model. Both models make suspicions about the way of nosy action that might be located. This chapter discussed a grouping plan of intrusions depend upon their detection in system occasions and applies pattern matching systems to speak to and identify them.

We see the significance of network intrusion. Specifically we talked about in great part which one experiences while utilizing the accepted security. Utilization of accepted efforts to establish safety can have genuine repercussions in the advanced complication of the web and the pervasiveness and infiltration of web which make all the systems, and PCs partnered to it, more inclined to such assaults. In this data age, when the worth of information and data is discriminating, burglary and misuse of information and data is the top necessity of all connections. In this manner, the necessity of advanced network intrusion identification system has taken prime imperativeness.

Additionally in this chapter, we had seen that general security is lacking for flow edge frameworks. Hence, Network Intrusion Detection System can into vicinity and all present corporate present fitting Network Intrusion Detection systems to guarantee the data in their framework. We moreover discussed the diverse occupations of a network intrusion detection system. From the examination, it could be deduced that a network intrusion detection structure could fill the gap by standard security frameworks like firewalls.

CHAPTER 3
BACKGROUND AND RELATED WORK

This chapter presents the background information and related work for this thesis. In particular, the chapter revolves around the discussion of various network intrusion detection systems of early age. None of these systems use pattern matching directly to represent and detect intrusions. The structure of this chapter is arranged to include a brief introduction on IDS and its background, followed by the early types of IDS, comparison amongst these types and finally several identified short-comings of IDS.

3.1 Introduction

Intrusion detection systems (IDSs) are used to identify, measure, and report unauthorized or unapproved network activities so that appropriate actions may be taken to prevent any future damage. On the basis of information sources that they use, Intrusion detection systems can be categorized into two classes; network-based and host-based. Network Intrusion detection systems analyse network packets captured from a network segment, while host- based Intrusion detection systems such as Intrusion Detection Expert System examine audit trails or system calls generated by individual hosts [61].

Network Intrusion detection systems use software programs, called sensors, to collect network packets. Because for detection purposes raw packets cannot be used directly, several sensors have pre-processing units that transform the packets into a useful format. Network Intrusion detection systems are gaining popularity since more and more systems are connecting over networks.

Intrusion detection systems can also be categorized according to the detection approaches they use. Primarily, there are two main detection approaches. They are as below:

- Anomaly detection
- Misuse detection.

The main difference between these two methods is that the anomaly detection approach analyses the properties of normal behaviour while misuse detection approach identifies intrusions based on features of known attacks.

There are several Intrusion detection system techniques for both anomaly and misuse intrusion detection. The techniques employed in these systems to detect anomalies are varied. There are some techniques based on of predicting future patterns of behaviour utilizing patterns, while others rely mainly on statistical approaches to verify anomalous behaviour. In both cases, observed behaviour that does not match expected behaviour is flagged because an intrusion might be detected. The major techniques used for misuse detection comprise model-based reasoning systems, expert systems, state transition analysis and keystroke monitoring. The accompanying subsections demonstrate further the two identification approaches as discussed above.

3.2 Anomaly Intrusion Detection

Anomaly Intrusion Detection is based on the normal behaviour of a subject (e.g. a system or a user). Any action that considerably deviates from the normal behaviour is considered as intrusive. That implies on the off chance that we could make an unremarkable action profile for a framework, then we can signal all framework states differing from made profile. There is a paramount distinction between anomaly based and misuse based technique. Anomaly based endeavour to detect the compliment of bad behaviour and misuse based detection system endeavour to apperceive the bad behaviour.

In this case, there are two possibilities:

False Positive: The anomalous activities that are not intrusive but are flagged as intrusive.

False Negative: The anomalous activities that are intrusive but are flagged as non-intrusive.

Anomaly detection system’s block diagram is as following:

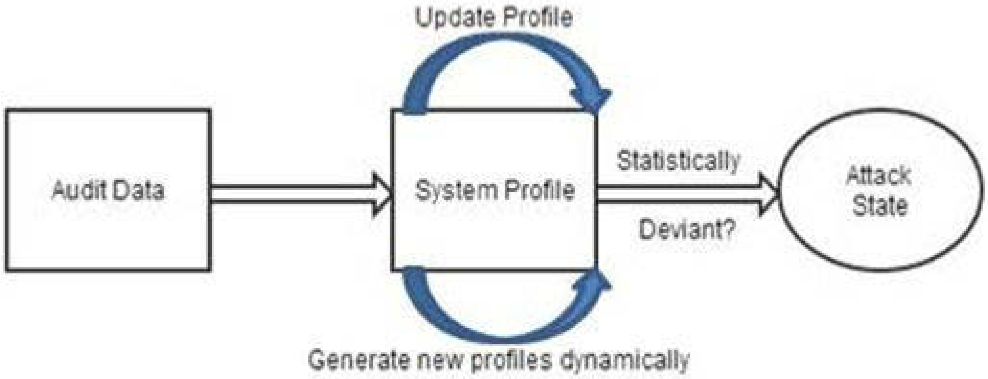


***Figure 3.1*- A Typical Anomaly Detection System**

The diagram shows, Anomaly detection surmises that intrusions are anomalies that obligatorily differ from normal behaviour [62]. Fundamentally, anomaly detection establishes a profile for normal operation, and signatures the activities that diverge considerably from the profile as attacks. The key playing point of anomaly detection is that it can identify obscure assaults. On the other hand, this focal point is paid for as far as a high invalid positive rate in light of the fact that in practice aberrances are not basically intrusive. Moreover, anomaly detection cannot detect the assailment that does not conspicuously deviate from normal activities. As the number of incipient attacks increases quickly, it is tough for a misuse detection approach to maintain a high detection rate. In integration, modelling attacks is a highly eligible and time-consuming job that leads to a heftily ponderous workload of maintaining the signature database [62].

The vital premise of anomaly intrusion detection is that intrusive activity is a subset of anomalous activity. This might seem plausible, considering that if an outsider breaks into a computer account with no notion of the compromised user's pattern of resource utilization, there is a good chance that his comportment will be anomalous.

Frequently, intrusive activity can be carried out as a sum of individual activities none of which is independently anomalous. Preferably, the set of anomalous activities is equipollent to the set of intrusive activities. Then, flagging all anomalous activities accurately reflects all intrusive activities, resulting in no erroneous positives or erroneous negatives. Though, intrusive activity does not always coincide with anomalous activity.

There are four possibilities, each with a non-zero probability:

i) Intrusive but not anomalous: These are erroneous negatives, or type I errors. However, the activity is intrusive but because it is not anomalous, we fail to detect it. These are called erroneous negatives because the Intrusion detection systems erroneously reports absence of intrusions.

ii) Not intrusive but anomalous: These are erroneous positives or type II errors. We report it as intrusive, as the activity is not intrusive but because it is anomalous. These are called erroneous positives because the Intrusion detection systems erroneously reports intrusions.

iii) Not intrusive and not anomalous: These are true negatives; the activity is neither intrusive nor reported as intrusive.

iv) Intrusive and anomalous: These are true positives; this activity is intrusive and is reported as it is additionally anomalous.

When erroneous negatives are not desirable, thresholds that describe an anomaly are set low. This results in many erroneous positives and detracts from the efficacy of automated mechanisms for intrusion detection. It produces additional burdens for the security officer as well, who must investigate each incident and discard many. Anomaly detectors incline to be computationally expensive because several metrics are often maintained that need to be updated against every system activity.

Anomaly detection methods that discover the intrusions through heuristic learning are relatively simple to maintain. A variety of techniques are utilized in anomaly detection. Predicated on their normal features, these techniques can mainly be categorized into four different paradigms: statistical based, feature based, predictive pattern based, and neural network based.

3.2.1 Statistical Technique

Statistical-predicated methods build operational profiles that describe normal behaviour of the monitored objects in a categorical time slot [63]. The normal profiles include the probability distributions of categorical measures such as the number of processes and the recollection utilization. Then, a statistical distribution profile of the audit data from an operating run of the monitored system is compared to the normal behaviour. If the two distributions are significantly different, a vigilant is raised. Intrusion Detection Expert System (IDES), Network Intrusion Detection Expert System (NIDES), discuss this method and its implementation.

The following, predicated on Network Intrusion Detection Expert System, the generic process of anomaly detection accommodates to illustrate, which is mostly statistical in nature. The activity of subjects and provokes profiles which are observed by anomaly detector represent their demeanour. These profiles are designed to utilize little recollection to store their internal state, and to be efficient to update because every profile may potentially be updated for every audit record.

As audit records are processed the system periodically engenders a value that is a quantification of the abnormality of the profile. And this is the value of function of the abnormality values of all the quantifications comprising the profile. Hence, if X_1, X_2,..., X_n represent the abnormality values of the profile measures Y_1, Y_2,..., Y_n respectively, and a higher value of X_i betokens more preponderant abnormality, an amalgamating function of the individual X values may be a squares of its slanted sum, the same as in

$$a_1 {X_1}^2 + a_2 {X_2}^2 + \ldots + a_n {X_n}^2 \quad , \qquad\qquad a_i \quad > 0\, , \tag{3.1}$$

where a_i reflects the relative weight of the metric Y_i; In broad, the quantifications Y_1, $Y2$, ., Y_n may not be mutually independent, and may require a more intricate function for cumulating them. Anomaly measures are just numbers without a well-defined theoretical substructure for coalescing them. For instance, utilizing multiplication of independent anomaly measures as a substratum of amalgamation is theoretically sound in likelihood computations but the relationship between anomaly measures and Bayesian likelihood numbers is not pellucid.

There are a number of types of measures encompass a profile, which include:

Activity Intensity Measures: These measure the rate at which activity is progressing. They are normally used to detect abnormalities in bursts of behaviour that might not be detected over longer term averages. An exemplar is the number of audit records processed for a user in one minute.

Audit Record Distribution Measures: This measure the distribution of all activity types in recent audit records. An illustration is the relative distribution of file accesses and Input / Output activity over the entire system usage for a particular user.

Categorical Measures: The distribution of a particular activity over categories is measured in this, such as the relative frequency of logins from the relative usage of each mailer, each physical location, compiler, shell and editor in the system.

Ordinal Measures: This measures activity whose outcome is a numeric value, such as the amount of CPU and Input / Output used by a particular user. While categorical measures count the number of times an activity occurred, ordinal measures compute statistics on the numerical value of the activity outcome.

The current comportment of each user is maintained in a profile. At customary intervals the current profile is merged with the stored profile. Anomalous demeanour is tenacious by comparing the current profile with the stored profile.

3.2.1.1 *Advantages and Disadvantages of Statistical Intrusion Detection*

The benefit of anomaly intrusion detection is that well studied techniques in statistics can often be applied.

a) For instance, data points that lie beyond a multiple of the standard deviation on either side of the mean might be considered anomalous.

b) The fundamental of the absolute difference of two functions over time might also be used as an indicator of the deviation of one function with respect to the other.

Statistical Intrusion detection systems additionally have several disadvantages:

a) Statistical measures are callous to the order of occurrence of events. That is, a pristinely statistical Intrusion detection system may miss intrusions that are betokened by sequential interrelationships among events.

b) Statistical Intrusion detection systems can be qualified gradually to a point where comportment once regarded eccentric, is considered normal. Intruders who know, that they are being monitored by anomaly detectors can qualified such systems. Therefore, most subsisting intrusion detection schemes cumulate both a statistical part to quantify aberration of deportment, and a misuse part that monitors the occurrence of concrete patterns of events.

c) It is arduous to determine thresholds above which an anomaly should be considered intrusive. Setting threshold too low results in erroneous positives and setting it too high results in erroneous negatives.

d) There is a constraint to the types of comportments that can be modelled utilizing pristinely statistical methods. The statistical techniques application to the formulation of anomalies requires the postulation that the underlying data emanates from a quasi-stationary process, a position that may not always hold.

More accurate models such as generalized Markov chains are more intricate and time consuming to build. It is difficult to determine thresholds above which an anomaly should be considered intrusive. Setting threshold too low results in false positives and setting it too high results in false negatives.

3.2.2 Feature Selection

A conundrum in anomaly intrusion detection is determining the correspondence between anomalous activity and intrusive activity. Given a set of heuristically culled measures that can have a bearing on detecting intrusions, the subset that accurately presages or relegates intrusions has to be tenacious. This is called feature cull. Determining the right measures is perplexed because the congruous subset of measures depends on the types of intrusions being detected. One set of measures will not likely be adequate for all types of intrusions. Predefined notions of the pertinence of particular measures to detect intrusions might miss intrusions unique to a picky atmosphere. A set of best measures for detecting intrusions must be resolute dynamically for best results.

Consider an initial list of *n* measures as potentially pertinent to prognosticating intrusions. The number of possible subsets of these n measures, which is the potency set of these quantifications, is *2n*. Because the search space is exponentially cognate to the number of measures, an exhaustive search for the optimal subset of measures is not efficient. Maccabe et al. present a genetic approach to probing through this space for the right subset of metrics [64]. Utilizing a cognition classifier scheme they engender an initial set of measures which is refined in the rule evaluation mode utilizing genetic operators of crossover and mutation. Subsets of the quantifications under consideration having low predictability of intrusions are weeded out and superseded by applying genetic operators to yield more vigorous measure subsets. The method surmises that cumulating higher predictability measure subsets sanctions probing the space of metrics more efficiently than other heuristic techniques.

3.2.3 Predictive Pattern Generation

The Predictive pattern generation is an anomaly detection technique that is predicated on the hypothesis, that sequences of events are not arbitrary but follow a discernible pattern. This result in more preponderant intrusion detection, because it takes into accounts the interrelationship and inductively authorizing among events.

The approach of time-predicated inductive generalization described by Teng and Chen [65] uses time-predicated rules that characterize the normal behaviour patterns of users [66]. The rules are engendered inductively, are modified dynamically during the learning phase and only good rules, e.g., rules with a high accuracy of presage and a high calibre of confidence remain in the system. A rule has high accuracy of prediction if it is correct most of the time, and it has a high level of confidence if it can be successfully applied many times in observed data. An instance of a rule engendered may be:

$$\mathrm{E1} \rightarrow \mathrm{E2} \rightarrow \mathrm{E3} \;\Rightarrow\; (E4 = 95\%, E5 = 5\%), \tag{3.2}$$

where *E1-E5* are security events.

This rule is based on previously observed data, which says that the pattern of observed events *E1* followed by *E2* followed by *E3* and the probability of seeing *E4* is 95% and that of *E5* is 5%.

This can generate more general rules that incorporate temporal relationships among events.

A set of rules engendered inductively by observing user behaviour encompasses the profile of the user. If the observed sequence of events matches the left hand side of a rule, a deviation is detected, but the following events deviate significantly from those presaged by the rule.

A major drawback of this approach is that unrecognized patterns of behaviour may not be apperceived as anomalous because they may not match the left hand side of any rule.

The strengths, claimed for this approach are:

a) A more preponderant handling of users with wide variances of demeanour but vigorous sequential patterns.
b) Ability to fixate on a few pertinent security events rather than the entire authenticates session that has been labelled suspicious.
c) More preponderant sensitivity to detection of contravention. Cheaters who endeavour to train the system during its learning phase can be discerned more clearly because of the semantics built into the rules.

3.2.4 Neural Networks

The fundamental approach here is to train the neural net on a sequence of information units [67] each of which may be at a more abstract level than an audit record. An input to the net consists of the existing command and the past w commands; where *w* is the size of the window of past commands that the neural net takes into account to predict the next command. Once the neural net is trained on a set of representative command sequences of a user, the net constitutes profile of the user and the fraction of incorrectly predicted next events then measures, in some logic, the variance of the user behaviour from his profile.

A notional diagram depicting the exercise of neural nets is shown in the following figure. The arrows directed at the input layer form the sequence of the last w commands issued by the user. Each input in this idealized demonstration encodes numerous values or levels, each of which exclusively identifies a command. Therefore the values of the inputs at the input layer communicate exactly to the sequence of the last *w* commands. The output layer conceptually consists of a single multi-level output that predicts the next command to be issued by the user.

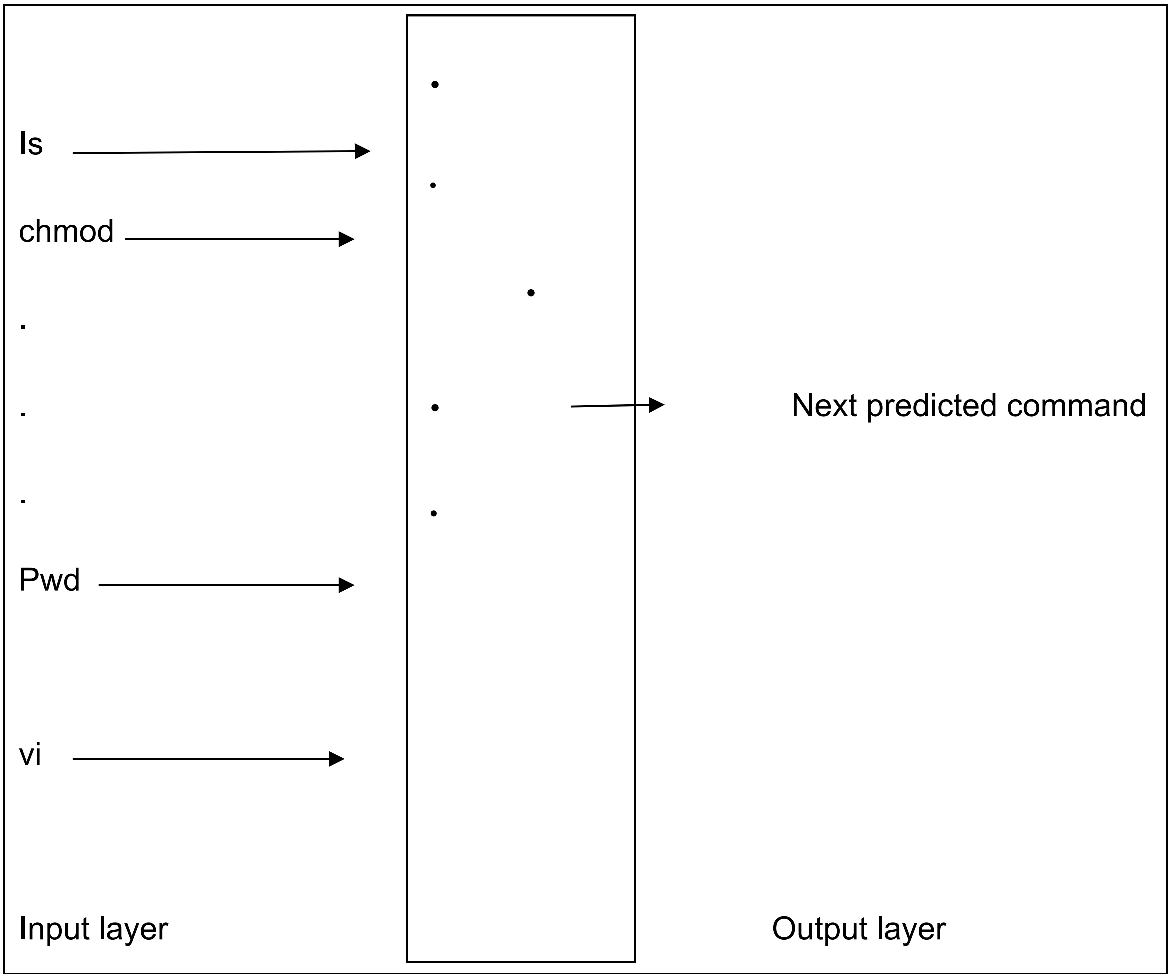


Figure 3.2 - **A Conceptual Use of Neural Nets in Intrusion Detection [68]**

For a good introduction to neural networks and learning in neural networks by previous propagation visually perceive the book by Winston [68],

The preferences of this methodology are:

a) The thriving of this methodology does not rely on any statistical propositions about the way of the underlying information.

b) Neural networks cope well with noisy data.

c) Neural networks can mechanically be considered responsible for relationships, between the sundry measures that influence the yield.

Following are the impairments of this methodology:

a) After noteworthy lapse and trial just, the topology of the net and the weights relegated to each component of the net are unflinching.

b) The volume of the window x is yet an alternate autonomous variable in the neural net configuration. In the event that x is situated in addition low, the net will do incapably, and on the off chance that it is situated further high, the net will persevere from insignificant information.

3.2.5 Bayesian Classification

Bayesian classification, portrayed by Cheeseman [69], is an unsupervised strategy of assignment of information. Its function by Autoclass [70], is to find for assemblies in the given information using Bayesian Statistical techniques. The doubtlessly procedures are attempts to focus by this procedure that provoke the information. The given information is not isolated into classes yet capacity decides the enrolment of every datum in probabilistic classes doubtlessly distinguished.

A few favourable circumstances of this methodology are:

a) In given information, Autoclass [70] mechanically evaluate the most conceivable number of classes.

b) There is no necessity of stopping runs, no specially appointed property measures and array criteria.

c) Uninterrupted and dissimilar qualities may be intense blend.

We are anxious with a downgrade of experiential conduct in statistical intrusion detection. The procedures that have been utilized work now have purposeful on regulated assignment in which client profiles are induced predicated on every client's watched appearance. The Bayesian downgrade technique might embrace the diligence of the most positive number of classes (foreseen processed), gathering clients with same profiles and subsequently consistent a characteristic downgrade of a set of clients.

This methodology is nascent and has not yet been actualized and tried in Intrusion identification frameworks. It is not self-evident, how well Auto class get a handle on characteristically sequential information like as a review trail, and to what degree will be at the empathy of statistical disseminations Auto-class managing review trails produced client. It is dubious to convey them if this strategy fits information through the Internet, whether any Auto class might be carried out in a slow decay at whatever point the beginning information are accessible, or else it relies on all information entered in the meantime. Being statistical in nature, it also suffers from some of the same generic failings of statistical systems, which relentlessness in moulding limits peculiarities right and the capability to impact client progressively conveyances class.

3.2.6 Belief Networks

Future frameworks may utilize Bayesian or other belief systems to amalgamate aberrance way. Bayesian frameworks embrace the show of causal conditions between subjective variables in graphical structure and regard the tally of the joint probability flow of the whimsical variables by designating one minute set of probabilities that relate simply to neighbouring centres [71]. This set embodies the previous probabilities of the whole root (centres without people) and the unforeseen probabilities of all the non-root centre points given all possible amalgamations of their prompt predecessors. Bayesian frameworks, which are Dags with bends addressing causal dependence between the gatekeeper and the tyke, sanction maintenance of affirmation when the characteristics of some irregular variables get kenned, and outfit a computational structure for evaluating the prohibitive characteristics of the remaining capricious variables, given this confirmation.

As an example, a trivial Bayesian system model of intrusion detection is shown in figure 3.3.

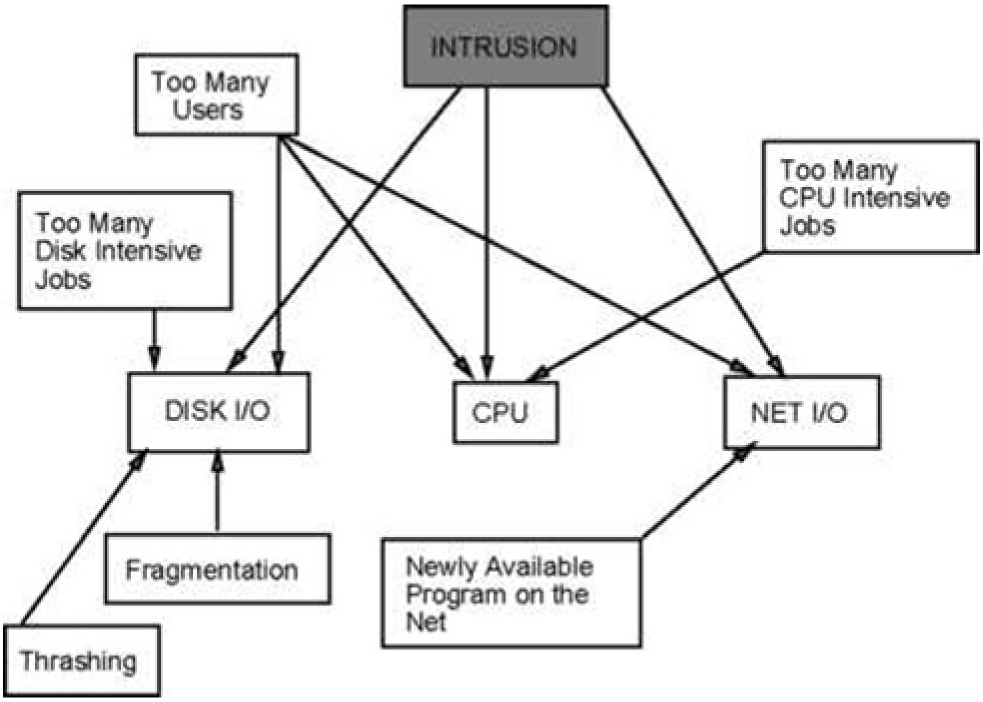

INTRUSION
Too Many Users
Too Many Disk Intensive Jobs
Too Many CPU Intensive Jobs
DISK I/O
CPU
NET I/O
Fragmentation
Thrashing
Newly Available Program on the Net

Figure 3.3 - **A Trivial Bayesian Network Modelling Intrusive Activity.**

Each one case speaks to a matched optional variable with qualities addressing either its conventional or atypical condition. When we can watch the characteristics of some of these variables, we can utilize Bayesian framework dissection to evaluate P (Intrusion/proof). On the other hand, when in doubt it is not picayune to evaluate the from the prior probability characteristics of the root centre points and the connection systems for every one guided round section. For an incredible exordium to Bayesian Networks, see the article by Charniak [72].

3.3 Misuse Intrusion Detection

Misuse intrusion detection suggests the disclosure of Intrusions by correctly segregating them early and apparently searching at for their occasion. Accurate methodology alone is not respectably extraordinary to recognize various sorts of intrusions due to their gateway and large schemas.

Intrusion imprints distribute the character, conditions, strategies and interrelationships amidst events that incite a break-in or other ill-use. Imprints are supplementary to distinguish intrusions and also attempted intrusions. An inadequate bliss of a signature may show an intrusion attempt.

A Misuse Intrusion Detector that just flags intrusions expected on the sample of data events translates that the state move of the machine achieves a bartered position when polished with impediment outline, paying little personality to the beginning state of the schema.

In this way, simply the designation of an intrusion signature without breaking the beginning state designation is now and again failing to offer an intrusion circumstance planarity gets. For a security model describing an intrusion and a case masterminded technique to its area, ostensibly watch it withal Gligor and Shieh [73].

Finally, we elucidate the different methodologies to misuse detection.

3.3.1 Using Conditional Probability to Predict Misuse Intrusions

This framework for determining interferences is identified with the one outlined out in past region except for that the "affirmation" is quickly a progression of external events rather than characteristics of aberrance measures. For ill-use interference revelation, we are fascinated with evaluating the prohibitive probability:

$$P(Intrusion|eventsequence). \tag{3.3}$$

Applying Bayes law as before to the above numerical explanation, we acquire:

$$P(Intrusion|eventsequence) = P(event\ sequence|Intrusion)\frac{P(Intrusion)}{P(Event\ Sample)}. \tag{3.4}$$

See as the yard arrangement of a foundation as the domain inside which the unexpected probability of intrusion is to be anticipated. A security expert joined with the offices wide framework may have the ability to process the previous probability of occasion of an intrusion on the grounds framework, or P (Intrusion), predicated on his experience. Additional, if the interference reports from the whole of the offices schemas are sorted out one can evaluate, for each one kind of event plan including an intrusion. It is $P(event\ sequence|Intrusion)$. The close repeat of happening of the event game plan in the entire intrusion set gives this probability. Besides, specified a set of intrusion free audit trails, one can compute, by examination and association, the probability $P(event\ sequence|Intrusion)$.

3.3.2 Production/Expert Systems in Intrusion Detection

The striking normal for utilizing engenderment schemas is the dissimilarity of control deduction from the specifying of the predicament result. A representation of the usage of these schemas in interference distinguish portrayed by Snapp and Samaha [74], this structure encodes understanding about ambushes and then implicative suggestion regulates in CLIPS [75] and insists realities contrasting with audit trail events. In if part, rules are encoded to specify the conditions requisite for an attack. The moment when all the conditions on the left side of a rule are satisfied, the actions on the right side are performed.

Convenient circumstances in the sufficient procurement of engenderment structures in intrusion disclosure are the critical measure of data to be dealt with and the typically absolutely requesting of the audit trail. The supervisor targets of engenderment frameworks in interference distinguish could be allocated into the going hand in hand with sorts: To interpret the occasion of a common intrusion on the backing of certain data. The essential issues in the use of taking care of framework/ expert are: It does not address the inbuilt trademark ask for or aggregating of the data. That is, the working memory segments (base effect) that match the left sides of productions to determine eligible rules for firing are not recognized by the system to be sequential. Moreover, the left a large portion of the era precept decides creation that the essentials joined with the AND association. To match the trademark course of action of the realities in this mapping, the Rete match technique impediments tests asked for every one set in the wake of qualifying that was made gatherings of work things identifying with the left a large portion of the arrangement [76].

Experience consolidated in the production/expert system is only commensurate to the security officer, who likes propensity, which may not be exhaustive. This is a viable thought; it is apparently stressed over the unlucky deficiency of a composed effort from security authorities in an attempt to focus their figuring out how to a complete security rule. Notwithstanding, if the tenet sets requirement to be planned and improved for the distinct situations, it may not be conceivable to go around this restriction. Just known vulnerabilities might be recognizing by this strategy.

There are concerns in programming building to look after the learning base. That is, should be included and erased from the guidelines in the guideline set progressions bring communications with whatever remains of the tenet set as a main priority.

To combine various intrusion measures and construct a cohesive picture of intrusions do uncertainty reasoning. The limitations of production systems that use uncertainty reasoning are well-known. See the book by Judea Pearl [71].

3.3.3 State Transition Analysis

In this methodology, taken in STAT and executed for UNIX in USTAT [77], attacks are represented as a sequence of state transitions of the monitored system. States in the example of the assault compare to the state framework and its co-partnered intelligent declarations that must be met to move to that state. The states are joined by progressive curves that speak to occasions needed to change the state. An assembled kind of occasions permitted in the model and does not need to compare one-to- unify with the review logs. It can distinguish strike designs with the goal that just the grouping of occasions is not allowed in ways more unpredictable than figuring out occasions. Besides, there is no component for open purposes to prune incomplete matches of different assaults through the necessities stressed in strong structure.

3.3.4 Keystroke Monitoring

This method utilizes keystrokes to figure out the time of the assault. The principal is to match the example of the particular succession of keys that indicate ambush. The hindrance of this methodology is the absence of dependable systems to catch the client's console without the backing of the working framework and techniques for innumerable of communicating the same level of assault on the keystroke. Also, without a semantic dissection of keystrokes, and nom de plumes furnished in client shells like Korn shell [78] can effortlessly crush this procedure. Client login shells frequently give the office of joining the names of shorthand parameters to the order arrangement. These are called monikers and like macro definitions. Since this method just breaks down the keystrokes and cannot recognize the computerized assaults that are the outcomes of the execution of a malevolent system.

3.3.5 Model-Based Intrusion Detection

This methodology was prescribed by Garvey and Lunt [79][79] and a variety on the misuse of intrusion discovery that joins together models of misuse of rationale with confirmation to help decisions about the event of ill-use. There is a database of assault situations, each of which comprises of an arrangement of behaviours that constitute attack. At any given moment, a subset of these attack scenarios are considered as the likely ones by which the system might currently be under attack. An endeavour is made to check these situations by looking for data in the review to demonstrate or discredit them called this procedure as anticipator [79]. And creates eager for the following set of behaviours to be confirmed in review, taking into account the models of the present animated, and passes these assemblies to the plan. Figure out how the plan is reflected in the conduct of the presumption of review information and translates it into a system dependent audit trail match. This should be the arrangement of conduct such action to be effortlessly recognized by the review, and must have a high likelihood of showing up during the demeanour. That is to say.

$$\frac{P(action|Behavior)}{P(action|\neg Behaviour)}. \tag{3.5}$$

As a map to a portion of the situations are heaping up, while others are falling catalogue is redesigned rundown of animated models. Rationale analytics in the verification in the heart of the framework permits one to overhaul the likelihood of ambush situations shows in the animated record.

The preferences of model-based intrusion detection are:

a) It is predicated on scientifically sound hypothesis of cerebrating in the vicinity of lack of determination. This is as opposed to a master framework methodology to manage lack of determination, where the decrease of the conclusions and the average is not as simple as proof of the opposite, it amasses. Master frameworks have additionally troublesome to clarify away the conclusions that repudiate affirmed later in the certainties. These issues can evade in the methodology to pondering the verification.

b) It can decrease a lot of transforming needed for each one review log through the checking of occasions coarser grained in latent mode and animated following of occasions finer-grained as occasions are discovered unpleasantness.

c) Representation plan furnishes autonomy from the underlying review representation.

The drawbacks of model-based intrusion detection are:

a) This methodology puts an unessential load on the individual inducing a manifestation of intrusion identification to designate numbers prove a principal and exact for the distinctive segments of the chart that speaks to the model.

b) It finished not demonstrate the productivity of the runtime of this methodology by building a model. It is not clear from the depiction of how the model might be gathered productively behaviours in the plan and this will have an effect on the conduct of the runtime to distinguish.

c) Intrusion detection based model does not supplant the piece of the statistical anomaly portion of intrusion detection systems, yet supplements it. For an extensive medication of the reasoning in the vicinity of questionable matter, see the book by Judea Pearl [71].

3.4 A Generic Intrusion Detection Model

A third type of Intrusion Detection approach was developed by Dorothy Denning, in 1987. She created a model for intrusion detection which was free of the framework, information sorting and particular mediations to be checked [21]. A brief clarification of the nonexclusive model is accommodating in the bond particular cases of intrusion detection systems exhibited in the past areas of the model and shows how these frameworks fit inside or fortified. Model is still faultless to portray the structure of a number of the present regulations.

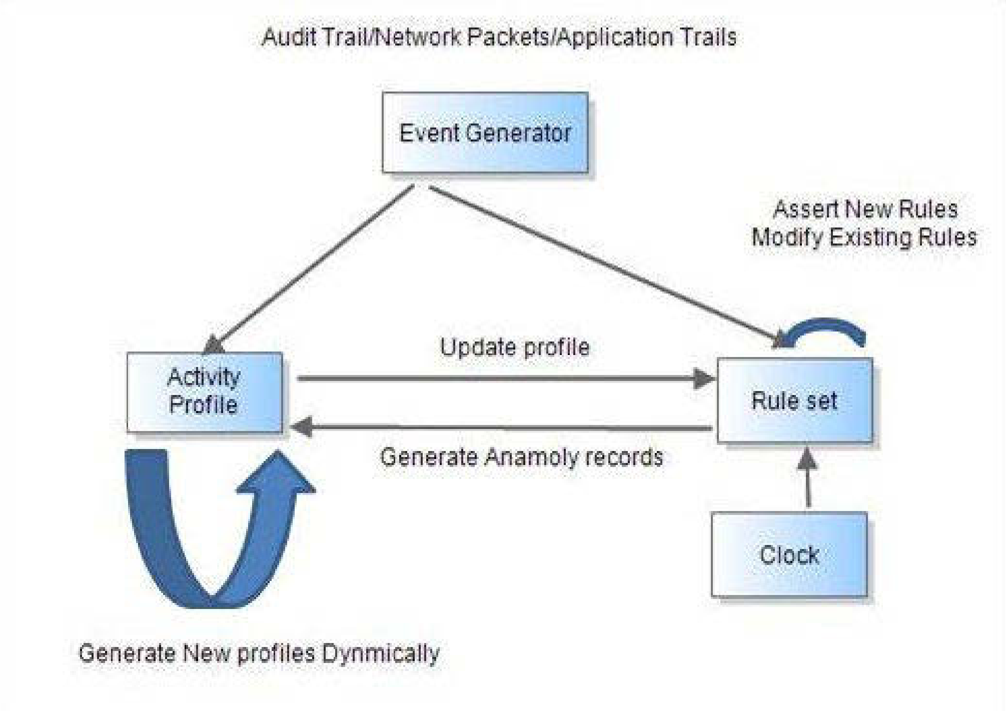


Figure 3.4 **- A Generic Intrusion Detection Model**

Figure 3.4 portrays the structural planning of the penetration in detection model. Generator of the occasion is non-specific; it may be the real occasions review logs, system bundles, or whatever available movement might be watched. These occasions serve as a foundation for the detection of a deformity in the framework. The Activity Profile is the global state of the intrusion detector. It holds variables that ascertain the conduct of the framework utilizing predefined measurable measures. These variables are magnificent variables, which are connected with every variable to figure out the style that an attempt to channel records the occasion. Matching records give information to overhaul its esteem.

For example, there may be **NumErrs** to the measure of statistical measure that computes the sum number of slips submitted by the subject in the logon session one. Every variable is joined with one of the statistical measures incorporated with the framework and is responsible for upgrading his condition on the support of the data held in the records matched the occasion.

The Activity Profile can also generate new profiles dynamically for newly created subjects and objects based on pattern templates. When you add new clients to the framework, or when new records is made, these templates instantiate new profiles for them. The Activity Profile can moreover incite abnormality records when some measurable variable undertakes a peculiar worth, for instance when Numbers assumes an extremely high esteem. The principle speaks to a general surmising instrument, utilizes the occasion logs, and records of irregularities, lapse dates and time, in addition to everything else, to control the movement of different parts to overhaul their state. Denning [21], however, utilizes a framework dependent upon the principle of surmising system to illustrate the way of the connection with different segments.

3.5 Comparison with Other Systems

The principle contrasts between the general model depicted above and the genuine frameworks portrayed in the past areas are:

- How are the guidelines of an aggregation comprising of the base.
- Whether the rule set is coded *a priori* or if it can adapt and modify itself depending on the type of intrusions.
- The nature of the connection between the dynamic profile and the Rule set.

The underlying topic, however the plans of statistical measures to distinguish great entrances, registering its esteem, and distinguish of oddities in the qualities indicated in most implanted frameworks in this way. In principle, the unit Profile last discovers peculiarities, while the unit performs a Rule set for misuse identification. Systems and diverse strategies could be substituted for these units without changing the theoretical perspective to a substantial degree.

On the other hand, some current strategies for the detection of oddity don't guide well in the inner parts of the last document. For example, the neural net methodologies to recognize aberrance do not fit effortlessly inside the system of savvy variables and record number to the quality of abnormalities. Not like learning and accommodation of tenet sets and characteristics well. It is additionally not clear in any unit TIM [66] will be set. TIM [66] discovers conduct inconsistencies, and along these lines may be a hopeful to be put in a profile last, yet it does so by producing manages and shoots them when conditions in the occasion of fulfilled some piece of the framework, which makes him a competitor for being likewise some piece of the Rule set. Extremely cutting edge approach, for example, model -based methodology is altogether different to fit into this system specifically.

3.6 Shortcomings of Current Intrusion Detection Systems

The following is an interpretation on the shortcomings of intrusion detection systems overall. Distinctive provisions rate an alternate path as per these topics of examination.

No Generic Building Methodology: In wide range, the expense of building the intrusion detection system starting with no outside help expansive is significant. This is a result of the absence of a precise methodology to building these frameworks. Popping any of these plans organizes the field itself. This may be halfway a consequence of the absence of a normal concession to systems to distinguish intrusions and part of the way on account of intrusion detection is a minor field of study, launched by Anderson in 1980 [27].

Efficiency: Systems have frequently endeavoured to recognize each possible intrusion and have not completed well in practice. Distinguish peculiarities; case in point, is computationally unreasonable in light of the fact that all arrangements supported by the framework may need to be upgraded for every occasion. Misuse identification has usually been actualized ordinarily by utilizing master framework shells that encode and match the signatures. These shells are frequently translating the standard set, and in this manner have a high time overhead. Besides, the principle sets permit just aberrant determination of the interrelationships between back to back occasions.

Portability: Intrusion detection systems have up to this point been indicted for single situations and have demonstrated laborious to use in different situations that may have homogeneous approaches and concerns. Case in point, moving the detection apparatus from a framework that furnishes an optional access control to a multi-level secure framework is not undeniable notwithstanding the way that the same concerns may apply to both. This is since a significant part of the framework may have a tendency to be particular to nature's turf being observed. Every framework is, in some sense, impromptu and specially crafted for its target. Reuse and retargeting is troublesome unless the framework plan in a general manner that it may be ineffectual or constrained force.

Upgradability: It is complicated to retrofit existing frameworks with original and best innovation to distinguish when they get accessible. Case in point, incorporating a Bayesian belief network to anticipate the intrusions into an existing system would be difficult on account of the absence of a reasonable comprehension of how this capacity must co-function with whatever remains of the framework.

Maintenance: The maintenance intrusion detection frameworks frequently require the abilities of incredible information of the most flexible security. Guideline sets overhauling, for example, regularly requires particular information about dialect master framework govern and see how the framework controls the tenets. This serves to maintain a strategic distance from undesirable co operations between decides that as of recently exist in the framework and those that are, no doubt included. Comparable contemplations apply statistical measures to add to the measurable component of the identifier.

Execution and Coverage Benchmarks: There is no information that has been distributed to date that quantifies the execution of Intrusion detection frameworks for a reasonable set of information defencelessness and nature's domain. Additionally, there is no distributed information on the scope of any framework or business or exploration. The information demonstrate that the rate of scope of leaps forward that will identify the framework in a nature. Merchants regularly treat scope qualitatively. This is halfway in light of the fact that it is challenging to discover the careful sorts of badgering and the recurrence of event in extensive situations, particularly the Internet. Be that as it may, there is no distributed information to conceal shortcomings in general society space.

No Good Way to Test: There is no effortless approach to test Intrusion Detection Systems. Upgrading rule sets, for example, often requires specialized knowledge about the expert system rule language and an understanding of how the system manipulates the rules. This helps avoid undesirable interactions between the rules already present in the system and those being added. Similar considerations apply to the addition of statistical metrics to the statistical component of the detector.

3.7 Summary of Intrusion Detection Techniques

There are a few intrusion detection systems that has been proposed and actualized. Most of them derive from the statistical intrusion detection model of Dorothy Denning. Some of them, case in point NIDX, sheaf, IDES, Midas, and the knowledge and the feeling of and CMDS utilize the review that was made by C2 or higher appraised machine, to enter. Others, for example Nice and NSM endeavour to dissect the mediations through the examination of the interchanges system and the stream of data in the system. For example, DIDS has stretched the extent of exposure through the dissemination organize oddity detection crosswise over heterogeneous and incorporated examination of halfway comes about because of these sources are appropriated for the identification of potential intercessions that may be missed by breaking down every individual source. Around non-measurable methodology to intrusion detection is to work by Teng which investigates a singular client review trails and endeavours to derive the relations between successive occasions, and neural net modelling of conduct by Simonian.

The methodologies of anomaly intrusion detection depend upon the dialect to speak to and discover intrusions, for example, ASAX, the advancement and provision customizing interface, a set of capacity calls library used to speak to and distinguish intrusions, as is the situation in STALKER, master frameworks, for example, MIDAS and NIDX, and elevated amount state machines for encoding and signatures match, for example, STAT and USTAT.

Bayesian Classification may include in a guaranteeing methodology for future intrusion detection frameworks, right now executed in Auto class. Depicted the review furthest reaches of the garage and scanning through Wetmore and Moitra, while being examined procedure to recognize examples of non-parametric by Lankewicz, permit the procedures to lessen review trail information pressure review in harshness, and occasions more unique that may be addressed at a later date by a security officer to recover data rapidly and proficiently. Non-parametric strategies to locate anomalies the focal point that they make no suspicions about the statistical dispersion of the underlying information, and convenient when these suppositions do not hold.

Intrusion detection can be regarded as a binary relegation quandary since it aims at distinguishing between normal and intrusion behaviour. A variety of relegation techniques rooted in machine learning or data mining technology can be employed to detect intrusions. In these approaches, prepared data labelled as either “normal” or “abnormal” is utilized for training a cognition algorithm which builds a detection model (profile) to prognosticate maleficent endeavours. Machine learning strategies are suitable to intrusion detection for the accompanying reasons. First and foremost, Machine learning systems distil the knowledge straightforwardly from authentic information. In this way, they oblige no manual work to concentrate former education.

Secondly, they can draw designs over fragmented information. Thirdly, it can speak to Machine Learning systems and the intelligence in an exceedingly conceptual manner, a capability that makes them well suited for taking care of a considerable measure of information. Variants of Machine Learning methods, for example, Support Vector Machines (SVM), genetic algorithms, decision trees and neural systems are utilized in intrusion detection.

CHAPTER 4
MACHINE LEARNING APPROACHES

This chapter will discuss what machine learning is and why it is used for specific classification tasks. After this discussion, the differences between supervised and unsupervised learning will be given. For both supervised and unsupervised learning, totally different algorithms and their characteristics are mentioned.

4.1 Introduction

When a workstation has to perform a particular task, a programmer's answer is to write down a computer program that performs the task. A computer program could be a piece of code that instructs the PC which actions are required to perform that specific task.

The field of machine learning deals with the higher-level question in a way to construct computer programs that automatically learn with expertise [80]. A computer program is alleged to learn from expertise *E* with relation to some category of tasks *T* and performance measure *P*, if its performance at tasks *T*, as measured by *P*, improves with expertise. Thus, machine learning algorithms inevitably extract information from computer readable information [81]. In machine learning, learning algorithms (learners) conceive to automatically distil information from example knowledge. This information may be used to create predictions concerning original data within the future and to supply insight into nature of the target concepts [81].

Applied to the examination at dispense, this entails that a workstation would be trained to classify alerts into incidents and non-incidents (task *T*). An attainable presentation measures (*P*) for this task would be the accuracy with that the machine learning program classifies the instances properly. The coaching experiences (*E*) might be tagged instances.

Machine learning could be a comparatively undeveloped scientific field that emerged from disciplines like computer science, applied math, statistics and biology, simply to call some [82]. Therefore, it is troublesome to state a transparent origin in terms of a selected scientific domain. Several authors and researchers with totally different backgrounds contributed necessary fragments to the current discipline we have a tendency to 0currently have. A problematic side-effect of this convergence is that the quantity of various nomenclatures and vocabularies [82].

A deeper insight into the fundamentals of machine learning is provided by the subsequent sections. Section 4.1.1 starts by giving an inexpensive definition of machine learning. This half is followed by sections that introduce the terms supervised and unsupervised learning, respectively.

4.1.1 Definition

A broad definition of the act of learning would not simply be supported engineering science. Rather, it would conjointly need issues of how humans and animals learn. This involves insights from disciplines like neurobiology and scientific discipline. Since this can be so much on the far side the main target of this thesis, the subsequent definitions and explanations solely target machine learning.

Nilsson describes the fundamental plan behind machine learning with the subsequent words: [82].

As regards machines, we would say, very broadly, that a machine learns whenever it changes its structure, program, or knowledge (based on its inputs or in response to external information) in such a way that it is expected future performance improves.

Machine learning and probably human learning can also be observed because the ability of generalisation. A system capable of learning is ready to generalise after being trained by some classes of samples. A system can, as an example, learn the looks of apples by being shown many totally different apples. The flexibility of generalisation allows the system to properly determine antecedent unseen apples.

Generalisation could be a powerful methodology but can easily go wrong. By being shown only green apples, the system described above might misleadingly generalise that all apples are green. Afterwards, the system would not determine associate unknown red apple as associate apple. So sufficiently scattered “learning apples” are necessary so that the system is able to learn the appearance of apples by being shown as many different apples as possible. Valid and fairly smart generalisations for the aim of recognising apples square measure that they are globular generally have a stalk and square measure sometimes red, yellow or inexperienced. By creating use of those easy rules, one may properly recognise most existing apples.

4.1.2 Formal Description and Terminology

This section tries to clarify the task of machine learning by using formal words. Additionally, a terminology is projected that is employed extensively within the remainder of this thesis.

As already mentioned, generally speaking, machine learning is the task of learning from examples. The available examples are organised in the form of a *training set* which is referred to as *X*. The precise components of *X* square measure known as training (coaching) samples and square measure outlined as tuples of the shape (X_j, C_j) wherever *C* refers to the set of separate category labels.

A machine learning formula is currently used to build a hypothesis out of *X*. The hypothesis could be operating of the shape $h\!: T \rightarrow C$ wherever *T* refers to the testing set.

Just as the training (coaching) set, the testing set is formed from samples of the shape $T_i,\ldots,T_m$ however the class label as within the training set is rather unknown. It is the hypotheses task to predict the class or in alternative words to classify samples.

So, every sample T_i is mapped to a separate class label C_j by the hypothesis *h*. For the aim of simplicity, let

$$C = \{1, -1\}, \text{ thus } \mathrm{h} : \mathrm{T} \rightarrow \{1, -1\} \; . \tag{4.1}$$

The equation $h(T_i) = 1$, would denote that the sample T_i of the testing set *T* was foreseen to be a member of sophistication class 1.

Fundamentally, *h* is an approximation to the target function known as *t*.

The target function $t : T \rightarrow C$ is outlined because it is function that predicts all samples of the testing set properly. Thus, within the scope of this definition, machine learning is the endeavour of approximating *h* to *t* nearly as good as attainable.

4.1.3 Application

So far it has not been stated why or when machine learning methods should be used instead of classical algorithms to solve a computational problem.

Machine learning becomes attention-grabbing once certain issues cannot easily be solved by using conventional algorithms. Aside from overflowing complexity, this can be the case with issues which might simply be delineate by stating examples than by developing algorithms [82]. A straightforward example is music classification, i.e., distributing a music track to a selected genre. Everybody is ready to explain a selected musical genre by enumerating several representative songs. But it is disproportionately harder to formally define music.

Machine learning is of nice facilitating in several fields. In medicine, machine learning conduct four-sided figure measure used to improve medical designation, in economy they fight to predict exchange costs to some extent and character and speech recognition would conjointly not work while the theoretical foundation was not provided by machine learning.

More relevant for this thesis is the area of computer security. Here, among other applications, machine learning is employed to notice malicious and abnormal activities in workstation networks and systems.

4.1.4 Benefits of Machine Learning

In specific, machine learning plays an important role within the following three areas of software system engineering [80]:

a) Data mining issues wherever massive databases might contain valuable implicit regularities that may be discovered automatically.

b) Complicated to program appliances, which are troublesome for ancient manual programming.

c) Software appliances that modify individual user’s predilections, akin to customized advertising.

There are several reasons why machine-learning plays an important role in these three domains.

First of all, for the classification of security events, an enormous quantity of facts of information has got to be analysed containing historical data, as was mentioned earlier. It is troublesome for mortals to search out a pattern in such a colossal quantity in sequence. Though, Machine learning, seems well-suited to overcome this problem and may therefore be able to discover those patterns.

With relation to the difficult-to-program applications, associate analyst's information is commonly implicit, and also the environments are dynamic [83]. As a consequence, it is very hard to program IDS using ordinary programming languages that require the exploitation and formalisation of knowledge. The adaptation and dynamic nature of machine-learning makes it an acceptable answer for this example.

Third, the environment of an IDS and its classification task extremely rely upon personal preferences. What could appear to be an occurrence in one surroundings could also be traditional in alternative environments [80]. This way, the flexibility of computers to learn allows them to grasp someone's "personal" (or organisational) predilection, and progress the presentation of the IDS, for these particular surroundings.

4.2 Supervised Versus Unsupervised Learning

Machine learning is divided into two categories; supervised and unsupervised machine learning algorithms [84].

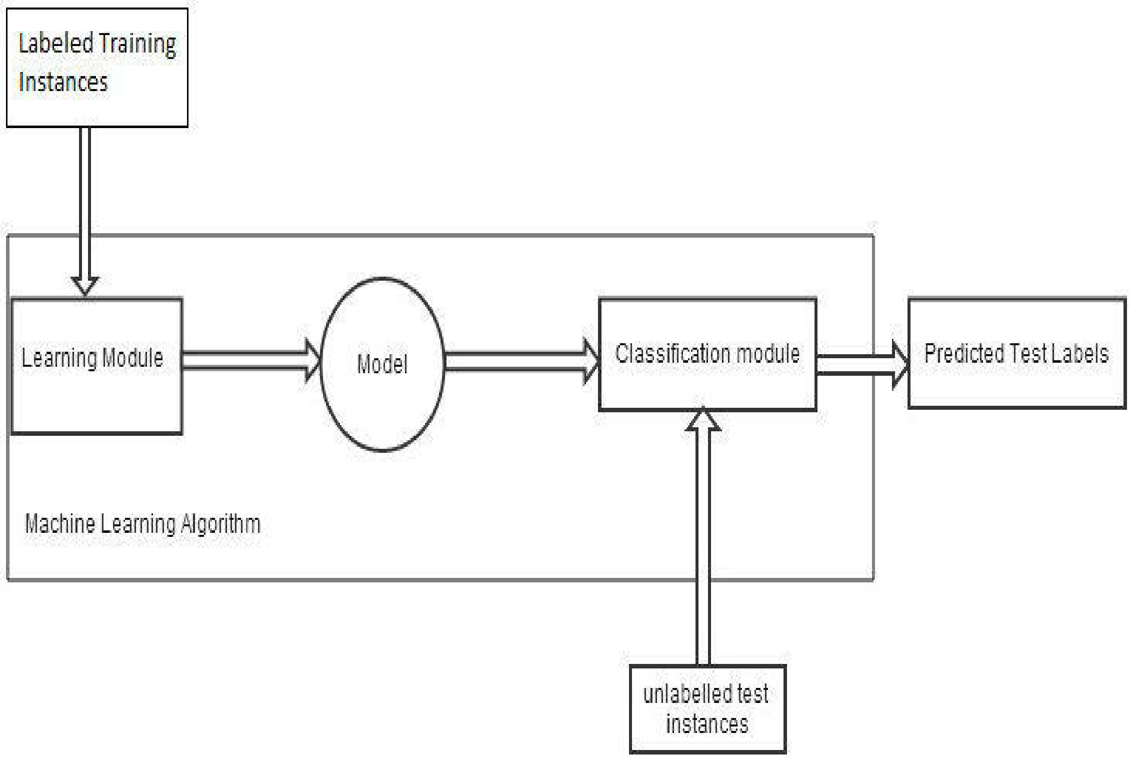

Labeled Training Instances
Learning Module
Model
Classification module
Predicted Test Labels
Machine Learning Algorithm
unlabelled test instances

***Figure 4.1*- Machine Learning Algorithm, supported [84].**

In supervised learning, the input of the training formula consists of examples in the shape of feature vectors with a label appointed to them. The target of supervised learning is to learn to assign correct labels to new unseen samples of constant task.

As revealed in Figure 4.1, a supervised machine learning formula consists of three parts: a model and a classification module, a learning module. The learning module builds a model supported a tagged training set. This model consists of a manoeuvre that is designed by the training module, and contains a group of associative mappings (e.g. rules). These mappings, once applied to associate untagged check instance, predict labels of the check set. The prediction of the labels of the test set is done by using the classification module.

In distinction to supervised learning, in unsupervised learning the machine merely receives inputs, however obtains neither supervised objective yields, nor plunder as of its atmosphere. Unsupervised algorithms learn from untagged examples. Unsupervised learning may be thought of as finding patterns within the knowledge and on the far side what would be thought-about pure unstructured noise [85]. The target of unsupervised learning could also be to cluster examples along the premise of their similarity [84].

4.3 Supervised Learning

The term supervised learning is a branch of machine learning which may even be understood as "learning by examples". This is equipped for the system machine learning has been outline up to presently. Nonetheless, it is not the sole approach to machine learning. Another necessary domain is named unsupervised learning and is roofed in Section 4.4.

The word supervised thereby refers to the method the training happens. Namely the supervisor (e.g. a workstation scientist) provides the learning algorithm with training samples which are used for generalisation. One might say that the man of science supervises the training method.

This operation may be compared to the way humans learn. Children tend to imitate other people’s behaviour. But they are doing not just imitate it; they try to generalise from it. This implies they will show constant behaviour in similar things. Thus by look the behaviour of adults (the coaching samples), youngsters try and learn (build a hypothesis). The training is performed by implicitly mistreatment the human brain (the learning algorithm).

Figure 4.1 illustrates the method of supervised machine learning. Place to begin is that the coaching set which mixes coaching samples with the specified output values (e.g. category labels). This coaching set is fed into a learning formula. The algorithms duty is to generalise from the coaching set by building a hypothesis. This hypothesis describes the coaching set and may then be used to analyse unknown testing samples. The speculation may be as simple as a direct work as clarified inside the following two areas.

Well known machine learning algorithms which are based on the concept of supervised learning are Support Vector Machines, neural networks and decision trees [86].

The range of supervised learning may be extra part into order and relapses issues. Every mixed bag of issues square measure succinctly specified inside the following two areas.

A guiding set is utilized by machine learning technique to make a theory that could be a generalization of the training learning. This theory is then used to characterize up to now obscure information.

Classification

As discussed in previous section, in categorisation issues the "yield" of the speculation could be a separate classification signatures *B&B* wherever *B* is that the situated of all class signatures of the honing set. A classification name is generally said even as class.

So a characterization disservice needs the mapping from a testing specimen to no less than one of numerous classifications $B = (B_i, B_2, \ldots, B_n)$. The characterization disadvantage is charged to be two-class or paired if n = 2, and multiclass if *n > 2[87]*.

For instance, two-class problems are the classification of network traffic to either benign or malicious. The classifications universal or strange additionally are ordinarily utilized. On the inverse hand, optical character make out could be a multi-class weakness. Each letter from "*a*" to "*z*" and "*A*" to "*Z*" manufactures a segregate classification to that composed letters square measure mapped to. An alternate multi-class hindrance is that the distinguishing proof of system activity. System streams frequently fit in with no less than one of the numerous conventions like correspondences convention, DNS or SNMP. Since the port-based methodology to spot system activity is not horrendously dependable any more owing to expanding *P2P* (Peer to Peer)-movement, a multi-class classification technique may be used to focus system activity [87].

Figure 4.2 chooses a straight two-class grouping impairment in *R2*. Each reason inside the chart joins someone is stature (*X-hub*) and weight (*Y- hub*). Red rounds articulate to females although blue crosses articulate to guys.

The challenge of the classification problem now is to learn a hypothesis which separates male from female points. By this means new learning focuses may be ordered and designated to a class: either male or female. Accordingly by essentially knowing the height and weight, the theory will tell (or at least figure) whether the individual is male or ladylike. Throughout the years specialists thought of amazingly unobtrusive ways.

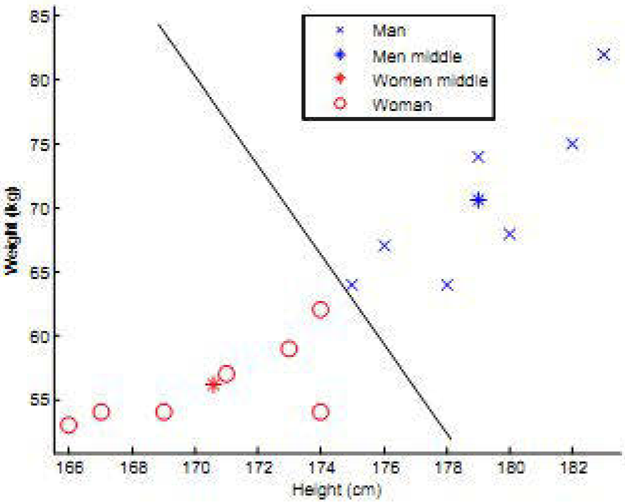


Figure 4.2 **- A linear two-class classification drawback [88].**

There are two diverse information conveyances, particularly male and ladylike focus. These two information sets may be directly divided in *R2* as represented by the dark line.

For grouping issues like neural systems and SVMs, this area is intended to supply exclusively an essential outline; the preparation strategy could be a simple direct classifier backed the Euclidian separation.

The vital illumination used to construct the speculation drops by scheming the mean purposes of every, the male (Y_m) and additionally the ladylike (Y_f) drilling specimens. In Figure 4.2 they are drawn as red and suffrage, respectively.

After the mean points of every disseminations square measure processed, the specific arrangement is proficient by critical whether a fresh out of the box new reason *X* lies closer to (Y_m) or (Y_f) to. This might be carried out by making utilization of a separation metric. For the point of this occurrence, the Euclidian separation is picked. Hence *X* fits in with the class to that the Euclidian separation is that the most diminutive? The theory acts as outline in Equation 4.2.

$$H(X) = B_m \quad \text{if } \ \mathrm{d}(\mathrm{X}, Y_m) < \mathrm{d}(\mathrm{X}, Y_f) \\ B_f \quad otherwise\ . \tag{4.2}$$

Regression

Contradicting to arrangement, the conclusion of any relapse issue is a persistent quality. In this way, the relapse theory is: *h*: *t* → *r*,

where *t* defines the testing set holding parts that square measure to be anticipated by the speculation.

The stock-market is an example for a typical regression problem. Dealers square measure ceaselessly inside the need of foreseeing the share trading system to exploit future occasions. Relapse calculations square measure regularly will not to deal with this sort weakness [89].

Figure 4.3 shows an average rectilinear relapse disservice. The graph holds ten blue crosses that speak to parts of the guiding set. Each honing example remains for an individual's weight and stature. One will see that each one crosses along almost sort a line in *R2*.

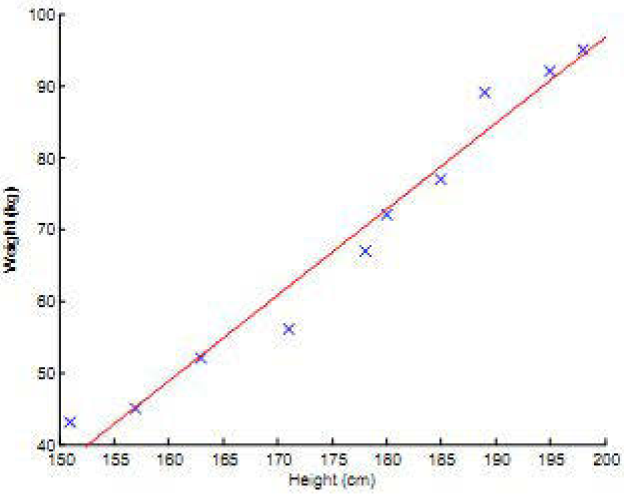


Figure 4.3 **- A rectilinear regression drawback.**

The conveyance of every last one of purposes of the guiding set may be approximated by a line in *R2* as represented by the line.

The objective of the relapse issue now is to make a theory which is primed to anticipate the heap of somebody by simply knowing the crest. In this way, the yield of the speculation could be a nonstop worth, particularly the heap. The theory is composed out of the training specimens recorded in Figure 4.3. Throughout this case, these preparation specimens are simply tuples (*a,b*); where *a* decides the data esteem (the weight) and *b* the yield esteem (the tallness). One of the ten tuples is (*163, 52*) that demonstrates that the individual's stature is 163 centimetres and likewise the weight is *52* kilograms.

Equation 4.3 (often called the linear model) indicates how the theory figures the ensuing persistent worth A out of the information vector B. The right part of the equation, $B^T\beta$ is written in matrix notation. The component β is named the consistent vector. This vector should be customized so with respect to the speculation to legitimately portray the instructing information set. Hence the reason for the learning technique is to take in cohort worthy consistent vector β.

$$\hat{A} = \widehat{\beta_0} + \sum_{j=1}^{p} b_j \widehat{\beta_J} = B^T \hat{\beta}. \tag{4.3}$$

There exist numerous algorithms for the determination of the constant vector. A well-known algorithm is the method of least-squares which is listed in Equation 4.4. RSS stands for the residual total of squares and denotes the total of the square error rates. Again, the matrix notation is given on the right as part of the equation. The lower the error rate, the higher the training knowledge set is matched by the hypothesis.

$$RSS\ (\beta) = \sum_{i=0}^{p} (y_i - {x_i}^T\beta)^2 = (y - X\beta)^T(y - X\beta). \tag{4.4}$$

The by-item with connection in this manner referred to as universal mathematical statements as noted in Equation 4.5 [90].

$$X^T(y - X\beta) = 0. \tag{4.5}$$

At last, Equation 4.6 means the response [90], i.e., the mathematical statement to unwind β.

$$\hat{\beta} = (X^T X)^{-1} X^T y. \tag{4.6}$$

As stated by Equation 4.5 the direct theory for the training information recorded in diagram 4.3 results in the line described by Equation 4.6.

The accompanying applies to every characterization still as relapse: a straight speculation is not generally the most suitable alternative to independent information occurrences. Distinctive preparing sets will require diverse speculations. For the purpose of effortlessness, a straight theory has been picked for this example.

4.3.1 Naïve Bayes

Naïve Bayes is an overseen learning technique. In controlled learning, the fact of the matter is to set up a schema to guide the data to yield; given the right values are provided by the head [9]. Gullible Bayes Classifier is reliant upon Bayesian Classification technique. Bayes rule learn the posterior probability *P(C|x)* using likelihood *P(x|C)* and prior *P(C)* with evidence *P(x)* as given below:

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}, \tag{4.7}$$

where *C* is the class and *x* is the data.

Naïve Bayes Classifier is a Bayesian Network which is needy upon the suspicion that all the insight qualities are prohibitively free given the target regard [9]. Given a plan of *n* attributes, the Naïve Bayes classifier makes *2n!* free suppositions [6]. It lessens a multivariate issue to an accumulation of univariate issue. In Naïve Bayes an alternate event is outfitted with a duple of *n* attribute values (*a1, a2, ..., a*), where *n* is the degree of data case.

$$\prod_i P\left(a_i | v_j\right) = P\left(a_1, a_2, \dots, a_n | v_j\right), \tag{4.8}$$

where v_j implies the yield quality transformed by the Naïve Bayes Classifier, v_j is the target regard that could be taken by the new event from the set V [1]. *P(*v_j*)* is the probability of target quality and *P(*$a_i|v_j$*)* is the unexpected probability that a particular feature f has attribute *a* given the target value *v*. In our case:

$$v = \begin{cases} 0 & connection = normal \\ 1 & connection = attack \end{cases} . \tag{4.9}$$

While planning, Naïve Bayes Classification obliges emerge scope of the acquaintanceship vector or data event. It does not need to be ready in distinctive cycles.

The Naïve Bayes classifier is expected for usage when features are independent of one and another inside each class, in any case it appears to work well in practice really when that opportunity supposition is not quality. It requests data in two steps:

- **Training Step:** Using the planning investigates, the framework evaluates the parameters of probability dissemination; tolerating qualities are prohibitively free given the class.
- **Prediction Venture:** For any unseen test illustration, the framework figures the back probability of that variety fitting within each one class. The technique then arranges the test specimen concurring the biggest back likelihood.

The class-restrictive autonomy suspicion significantly rearranges the preparation venture since you can gauge the one-dimensional class-contingent density for each one characteristic exclusively. While the class-restrictive freedom between characteristics is not accurate by and large, research indicates that this supposition works well in practice. This suspicion of class autonomy permits the Naïve Bayes classifier to better gauge the parameters needed for correct grouping while utilizing less preparing information than numerous different classifiers. This makes it especially viable for datasets holding numerous indicators or characteristics.

In testing stage, the restrictive probabilities that are found in preparing stage, are utilized for another occurrence of information. The Naïve Bayes technique is connected on the credits of the information to discover the yield esteem v_{nb} for each of the strike connection and ordinary association. The information is named as an attack or normal relying upon which connection sort v_{nb} is most extreme.

Naïve Bayes is the most useful learning technique. Its execution is tantamount to other neural system and decision trees in numerous spaces. Naïve Bayes systems are broadly utilized as a part of grouping of content issues. A fascinating contrast between Naïve Bayes and other neural system systems is that; in Naïve Bayes there is no need of unequivocal pursuit in space of theory [9]. It is framed by discovering the probabilities of different information fusions in training dataset.

The Naïve Bayes model may be a heavily simplified Bayesian probability model. During this model, take into account the probability of end results consequence given many connected proof variables. The probability of the top result's encoded within the model beside the probability of the proof variables occurring provided that the top result happens. The probability of end results proof variable provided that the top result happens is assumed to be freelance of the probability of different proof variables provided that the top result happens. Now we have a tendency to can take into account end results example.

Suppose that a hypothetic automotive alarm that responds properly ninety nine of the time. The opposite 1% is split into two groups, false constructive and false unconstructive. False positives (+) conjure all the things during which the automotive alarm burst, however wherever there is no criminal activity occurring. Assume that a hundred and twenty fifth of the time that the alarm rings, that this can be the case. False negatives conjure all of the things during which the automotive alarm does not burst, however there is End results tried felony. Assume that this event additionally makes up a hundred and twenty fifth of all cases during which the alarm does not burst. Now, assume that the probability of criminal activity occurring with this specific automotive to be a hundred and twenty fifth in any given hour. Over an amount of one hour, the automotive is left unsupervised. The alarm burst once during this time-what is that the probability that a felony occurred once the alarm went off? What is the probability that a felony did not occur once the alarm went off?

One way to approach this downside is to use the conception of natural frequencies. Natural frequencies translate the probability into concrete whole numbers before transferring them into chances as an example, a probability that a good coin provides heads is thought of because the concept out of one thousand cases, five hundred are heads.

Now, we will take into account the alarm example employing a Naïve Bayes classifier. Assume that, we have a group of examples that monitor some attributes like whether or not it is descending, whether or not End results earthquake has occurred, wherever the automotive is pose, etc. Let is assume that we have a tendency to additionally grasp, victimization this monitor, concerning the behaviour of the alarm beneath these conditions. In addition, having data of those attributes, we have a tendency to record whether or not or not a felony really occurred. We will take into account the class of whether or not a felony occurred or not because the category for the Naïve Bayes classifier. this can be the data that we have a tendency to have an interest in. the opposite attributes are thought-about as data which will offer United States proof that the felony has occurred (the actual quality of this data as proof are mentioned later).

The Naïve Bayes classifier functions on a robust independence assumption. This implies that the probability of one attribute does not have an effect on the probability of another. As an example, we have a tendency to assume that the probability of End results earthquake does not have an effect on the probability that the alarm burst. Therefore for 2 events X and Y, the probability of X occurring provided that Y happens is just the probability that X happens. In different words,

$$P\,(X\,/\,Y) = P\,(X). \tag{4.10}$$

The sturdy independence assumption is unreal in most cases. Given a series of *n* attributes, the Naïve Bayes classifier makes *2n!* independent assumptions. Even so, the results of the Naïve Bayes classifier are usually correct [91]. Following circumstances proves that the Naïve Bayes classifier performs well and why? They state that the error may be results of three factors: coaching knowledge noise, bias, and variance. Coaching knowledge noise will solely be reduced by selecting smart coaching knowledge. The coaching knowledge should be divided into numerous teams by the machine learning formula. Bias is that the error owing to groupings within the training data being large.

Variance is the error due to these groupings being too little. The error owing to bias in zero-one loss is explicit to be usually abundant not up to the error from variance.

In the training phase, the Naïve Bayes formula calculates the possibilities of a felony given a specific attribute and so stores this probability. This can be perennial for every attribute. The time taken on this activity is proportional to n, the amount of attributes, and also the quantity of your time taken to calculate the relevant chances for every attribute. The number of your time taken to calculate the probability of felony for end results attribute is proportional to the amount of examples given to calculate the probability from, and to the amount of various values that attribute will take and hence, to the number of your time taken to end the training section.

In the testing section, the number of your time taken to calculate the probability of the given category for every example within the worst case is proportional to n, the amount of attributes. Assumptive that the index *vi*, the worth of the attribute, is accessed in constant time, the time taken to seek out the relevant chances and multiply them along is *O(n)* for every example. Given e examples, the time taken is proportional to *O(ne).* As a result of it is usually true that $e >> n$, the time taken within the worst case is thus *O(e).* So the testing section takes an equivalent quantity of your time within the worst case because the coaching section.

4.3.1.1 Bayesian Decision Theory

Bayes theorem relates the conditional and "prior" (marginal) chances of two random events. This theory has played an important role in several applications, and might be explained by a discussion of playing cards. Suppose, $X\ (Y_1)$ and $X\ (Y_2)$ is the probability of drawing a red and black card respectively. Since there are solely red and black cards in every deck of playing cards, the probability of either occurring is precisely one.

Next, let $X\ (O)$ is the probability of drawing a picture card (a Jack, Queen or King). The chance $X(Y_{1|}O)$, i.e. the prospect of depiction a red picture card, will then be found with the following trivial probability calculation.

$$X(Y_1\, O) = \frac{X\ (O|\, Y_1)\ X\ (Y_1)}{X(O)}. \tag{4.11}$$

That is, the probability that the card is red provided that it is a face. Equally for black face cards:

$$X(Y_2\,O) = \frac{X\,(O|\,Y_2)\,X\,(Y_2)}{X(O)}. \tag{4.12}$$

For end results experiment with *n* reciprocally exclusive outcomes (as is often the case with text classification), the divisor is shown to equal one. Moreover, they are not reciprocally exclusive; it stays constant for every outcome, and might therefore be unseen from every calculation.

In Thomas Baye's own words: "If there be two subsequent events, the probability of the second b/n and the probability of both together P/N and it being first discovered that the second event has happened, from hence I guess that the first event has also happened, the probability in the right is P/b." [115]

Now, it is somewhat unlikely that any card player instantly would notice this terribly fascinating. He already knows this probability is *H*, since there are twelve face cards within the deck, and *1/2* them are of every colour. We are saying that the parameters are fastened, which the information is random (i.e. that card is drawn).

Let's flip these assumptions around, and take into account the parameters to be random, returning from a distribution of values, and also the knowledge to be notable. Say we are out fishing. In advance, the native (very mathematically inclined) fishermen teach United States that, at now of year, the probability of obtaining a cod is $X(Y_1)$ = half-hour, and for trout $X(Y_2)$ = two hundredth. We have a tendency to decision this previous data, or a priori probability.

Similarly, the fishermen have taught us a characteristic, or a *feature*, the two kinds of fish have: A cod usually has a distinct chin barbell, while the trout does not. In fact, the fishermen have found that, 90% of all cod fish have visible chin barbell, but only 5% of trouts do. We call this the *class- conditional probability* for that feature. Because the fishermen have been fishing all their lives, they've deducted these probabilities from the *known data* (i.e. the fish they've caught).

We are in luck, and get a bite on the primary throw. With the fish in our hands, we observe that it has a distinct chin barbell clearly visible. We use our newly learned information to determine what our catch is. How likely it is that it is a cod, based on the size of the chin barbell and the fact that it allegedly is 30% likely to catch a cod? How likely is it that it is a trout based on the same criteria? The priori and class-conditional probabilities for the two fish inserted into Baye's equation gives us:

$$X\left(\frac{cod}{barbell}\right) = X(barbeel\,|cod) * X\,(cod) = 90\% * 30\% = 27. \tag{4.13}$$

$$X\left(\frac{trout}{barbell}\right) = X(barbeel\,|cod) * X\,(trout) = 5\% * 20\% = 1\%. \tag{4.14}$$

Notice that, as mentioned earlier, the the right hand side denominator (*X (barbell*)) has been excluded from the calculations, as a result of it is merely a relentless in each.

The equation provides United States the posterior probability for every fish. They decide the fish class that offers the best price. They have currently classified their fish employing a Bayesian decision.

4.3.1.2 Naïve Bayes Classifier

A Naïve Bayes classifier is a classifier that adopts Bayes decision theory to choose a class that a knowledge item belongs to. As the fishing example within the previous section, the selections are probabilistic - or additional specifically supported most a posterior. The information is fed through the classifier, even as fish catch was evaluated in fanciful fish classifier that produces a posterior probability for every class. Finally, the class with highest probability is chosen for that specific knowledge item. Expressed as a statistical method, this turn into:

$$h_{N_B} = \underset{h_j \in H}{arg\,max}\; P(h_j) \prod_i P(O_i\,|\,h_j). \tag{4.15}$$

In the above formula, you will notice that the prior probability is multiplied with the merchandise of the many state-conditional chances. Within the fishing example, we have a tendency to only one state-conditional probability to multiply with - as a result of we have a tendency to solely one feature. In additional realistic applications, there will be many, maybe thousands of options to include. As an example, in a very text classifier, a typical feature set is just the frequency of the words occurring within the posts. However the state-conditional chance for the words is calculated is represented in next section.

This augmentation of each characteristic is state-restrictive likelihood intimates that we have made the thought that:

$$P(O_1, O_2, \dots O_n \mid h_j) = \prod_i P(O_i \mid h_j). \tag{4.16}$$

This is supposed to be the Naïve Bayes presumption and intimates that $O_1, O_2, \dots O_n$ are not totally independent. As an illustration, inside the instance of a content classifier, with each saying being a characteristic, the connections of the words are lost once they are dealt with as divide elements. This fraudulent presumption is wherever the Naïve Bayes classifier gets the "guileless" a piece of its name.

On the other hand, even in requisitions like content classifiers, the Naïve Bayes classifier demonstrates to figure amazingly well, disregarding its guilelessness. Truth be told, examination demonstrates that in a few cases, it performs similarly to some fundamentally extra muddled methods. Its high execution to quality size connection has been a pivotal recognizes the Naïve Bayes classifier's prosperity.

Merits: This system needs a low amount of training data to gauge the parameters vital for classification. The classifiers upheld this technique displayed high correctness and speed once connected to huge databases.

Demerits: This method works well furnishing accepted choices are free; once reliance emerges then it furnishes low execution.

4.3.2 Linear Discrimination Analysis (LDA)

Linear Discrimination, initially created in 1936 by R.A. Fisher, may be a variable procedure of order [92]. Segregation investigation closely resembles multivariate dissection aside from that the variable is absolute rather than consistent [93]. In Discrimination examination, the plan is to foresee classification enrolment of unique perceptions backed an aggregation of indicator variables. LDA generally makes an endeavour to search out direct combos of indicator variables that best divide the groups of perceptions. These combos are regarded as Discrimination capacities [94].

Assume there are K totally distinctive groups, each accepted to own a variable measurable dispersion with mean vectors μ_k $(k=1,...,k)$ and standard change lattice $\sum$ the specific mean vectors and difference lattices are about dependably obscure; the most likelihood evaluations are acclimated assessment these parameters.

The thought of LDA is to order perceptions commotion to the bunch k, that minimize the inside grouping transform, i.e.

$$k = \underset{k}{arg\ min}\ (x_i - \mu_k)^T \sum^{-1} (x_i - \mu_k). \tag{4.17}$$

Under variable accepted suppositions, this could be love discovering the group that amplifies the likelihood of the perception. By and large, we will gauge past likelihood exploitation the extent of the measure of perceptions in every bunch to the full.

As a sample, let$\pi_k = n_k/n$be the extent of group k, such that $\pi_1 + ... + \pi_k = 1$. At that point, as opposed to expanding the likelihood, the back likelihood is augmented; the perception has a place with a particular bunch,

$$k = \underset{k}{arg\ max}[-\tfrac{1}{2}(x_i - \mu_k)^T \sum^{-1} (x_i - \mu_k) + log\pi_k]. \tag{4.18}$$

Disentangling above, the k LDA capacities are,

$$d_k(x) = x^T \sum^{-1} \mu_k - \tfrac{1}{2} \mu_k{}^T \sum^{-1} \mu_k + log\pi_k. \tag{4.19}$$

The point when the thought of normal change lattice is not euphoric, a classified difference framework for each one bunch is utilized. These effects in Quadratic Discrimination Analysis (QDA) as the separating limits are quadratic bends instead of straight lines. Box is *M* test is utilized to test the homogeneity of fluctuation [95]. When the check is basic, QDA is utilized. QDA does not surety end effects enhanced characterization rate. In the binary case, two direct discrimination capacities are built as follow:

$$d_1(x) = x^T \sum^{-1} \mu_{1k} - \frac{1}{2} \mu_1{}^T \sum^{-1} \mu_1 + log\pi_1. \tag{4.20}$$

$$d_2(x) = x^T \Sigma^{-1} \mu_2 - \frac{1}{2} \mu_2{}^T \Sigma^{-1} \mu_2 + log\pi_2. \tag{4.21}$$

In the event that *$d_1(x) > d_2(x)$,* the perception x are allocated to bunch one, generally to group two. The two discrimination works likewise might be joined, i.e.,

$$\begin{aligned} d(x) &= d_1(x) — d_2(x) \\ &= x^T \Sigma^{-1} (\mu_1 - \mu_2) - \frac{1}{2}(\mu_1 + \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2) + log \ (\frac{\pi_1}{\pi_2}) \end{aligned} \tag{4.22}$$

In the event that $d(x) > 0$, the reconnaissance x are allocated to group one, generally to bunch two. The last two segments inside the comparison (4.20) are steady given a learning set; the discrimination work coefficients are $D = \sum^{-1}(\mu_1 - \mu_2)$. The coefficients reflect the joint commitment of the variables to the work, subsequently demonstrating the impact of each variable inside the vicinity of the others. The institutionalized constants $D^* = diag(\sum)d$ are figured by duplicating each coefficient by the quality deviation of the comparing variables. Once the variable scales disagree respectably, the institutionalized consistent vector furnishes higher information concerning the relative commitment of each variable to the standard discrimination work.

Assume there are two groups of *p* indicator variables, which allow for development of LDA capacities exploitation all indicators. A sensible technique is to settle on fundamental variables exploitation stepwise system, which uses the Wilks' Lambda statistics to identify significant independent variables of the discriminated functions [96]. The Wilks' Lambda model maximally separates between groups by expanding the variable *F* size connection inside the tests of varieties between bunches implies that.

Segregation capacities are built backed two presumptions, i.e., multi-typicality in every bunch and homogeneity of difference between groups. Assuming that there are some clear cut indicator variables, these two suspicions are typically degraded, which can impact the standard of the models and expectations. Diverse constraints with discrimination analysis are that the mean vectors of the groups should be discernable which the measure of perceptions in every group should be greater than the measurement of the variables. Assuming that the mean vectors do not appear to be totally distinctive enough; it is hard for LDA to yield tight grouping rates. In the event that the perceptions in a few groups are confined, a stepwise strategy is required to select potential fundamental variables before LDA is utilized.

While performing classification problem, the arrangement rate must be computable. One clear strategy is named re-substitution, which applies the Discrimination model to the first honing learning set to take a gander at the recurrence of appropriately grouped perceptions. Re-substitution typically overestimates the right order rate.

An alternate method for action the likelihood of right grouping is q-fold cross approval [97]. For q-fold cross-approval, the first specimen is parcelled off into alphabetic character subsamples. One subsample is looked after for acceptance of the model designed from the inverse q-1 subsamples on every event. The system is perpetual alphabetic character times, with each of the alphabetic character subsamples utilized particularly once for acceptance. The outcomes are joined to give one arrangement rate gauge. A chose provision is that the "leave-one-out" cross approval, wherever alphabetic character levels with the measure of perceptions inside the first learning set [98].

4.3.3 Artificial Neural Networks

An Artificial Neural Network (ANN), as described by [99], may be a network in which principal crucial component is named a 'neuron'. The nerve cell has connections with diverse neurons by which it gains and transmits learning. The nerve cell performs the ensuing calculation: the qualities of the connections into the nerve unit are expanded by the some weights of these connections:

Let y_k be the value sent over the k_{th} group

Let w_k be the lumber sent over the k_{th} connection

$$a_j = \sum_{k=1}^{p} w_k y_k \,. \tag{4.23}$$

At that point, a non-straight enactment capacity f is connected to the present worth:

$$y_j = f(a_j). \tag{4.24}$$

A regular decision for f is that the sigmoid capacity, $f(x) = \frac{1}{1 + e^{-x}}$. Notwithstanding, diverse choices for the actuation work encapsulate the tan work.

The neurons are sorted out into two extra layers. The learning experiences the layers in a direct manner, going first through the data layer, passing through every concealed layer progressively, work it is at long last yield by a definitive layer. Shrouded layers are the layers between the information and yield layers in an exceptionally encourage forward neural system [99].

The methodology of back-engendering is one procedure used by neural systems to discover the insight set [99]. The neural web first experiences forward spread thus contrasts its yield and the specific classes of the samples. It then alters the weights of each layer to rose take in the effects. Each segment of forward- spread End comes about back proliferation is named an age. A neural web keeps running ages work an exact edge of correctness is arrived at or an exact amount of ages is run, whichever comes first. Back engendering upgrades the weights of the shrouded and yield layer. This system is said in automation and remote control system [99].

4.3.4 Support Vector Machines (SVMs)

Support Vector Machines, conjointly referred to as SVM, may be a sort of learning machine that use supervised learning models to investigate and classify knowledge. The most popular area of usage is to construct associate optimum model which will distinguish new knowledge points into one amongst two totally different categories. Initial bestowed by Vapnik and Alan Jay Lerner in 1963 [100], the most thought is to construct a hyper plane because of the apparatus of the two categories. The main apprehension concerning hyper planes is that it can easily be applied in higher dimensions that make them ideal to general solutions.

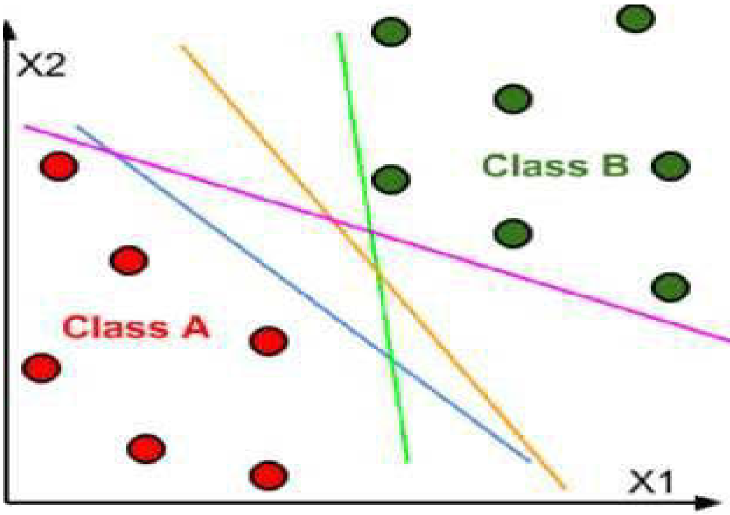


Figure 4.4 **- A group of labelled data points from the test data that are linearly separable. The data points have been separated by 4 hyper planes which will classify them correctly.**

When dealing with a linearly separable data set, there can be up to infinite ways to construct a hyper plane to correctly divide a data set into two classes as seen in figure 4.4. However, this is often where SVM stands out since the strategy bestowed by Vapnik can associate optimum hyper plane such as the one seen below in figure 4.5.

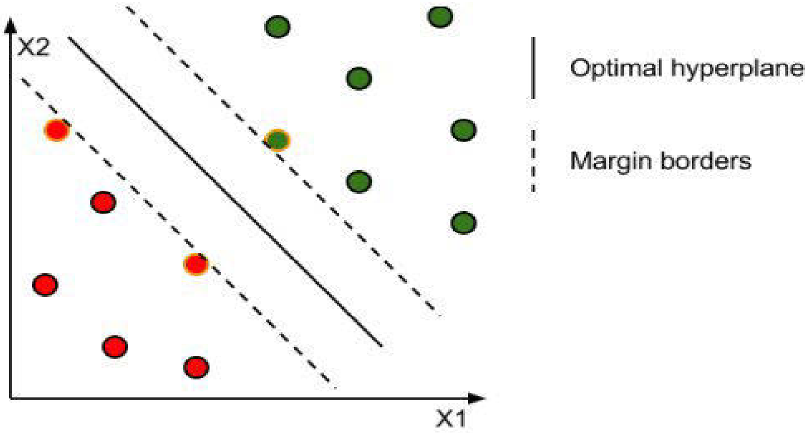


Figure 4.5 **- An optimum placement of the hyper plane that divides the 2 classes of the check knowledge whereas maximising the scale of the margin.**

It is referred to as the optimum hyper plane within the sense that it is created in order that the area between the purpose highest to the hyper plane and also the hyper plane itself is maximized; this distance is named the margin. Associate maximized margin can increase the chance of replacement information to be classified properly, albeit subjected to some noise. A more in-depth verify however the margin is calculated is seen in 4.5. SVM can convert the information points to m-dimensional vectors, and therefore the hyper plane created to divide them are going to be (m-1)-dimensional. Thanks to this, the information points that bit the maximized margin is named support vectors.

$$Minimize\frac{1}{2}\|w\|^2|\ \ subject\ to\ y_i(w^T x_i|b) \geq 1. \tag{4.25}$$

This remains a rather complicated downside since it is captivated with *w* however with the assistance of the Lagrangian Duality Theorem, this downside is simplified into functions of the support vectors instead. The new downside then becomes the subsequent as seen in equation 4.26. The optimum hyper plane will then be created from the answer victimization equation 4.27.

$$\sum a_i y_i x_i\ , \tag{4.26}$$

where the support vector is that the corresponding *xi* and *α* is larger than zero.

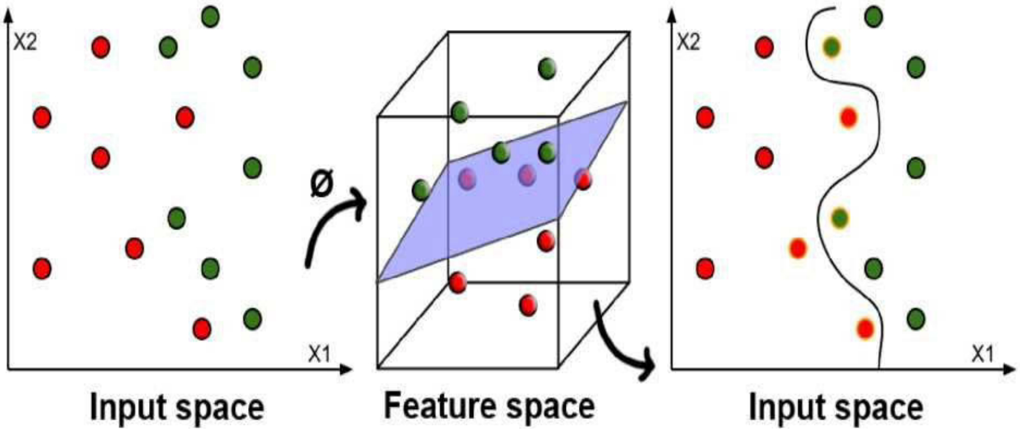


Figure 4.6 **- Left: A non-linearly severable knowledge set within the input area. Middle: an equivalent input in an exceedingly feature area wherever a linear classification is found. Right: The input within the feature area is then remodelled into the input space**

Not all data sets are linearly separable as seen in figure 4.6, in fact that is usually the case when dealing with real data. An artless trick is that the on top of algorithmic rule to maximise the margin still works in higher dimensions, thus if you are featured with non-linear severable points within the x area, you will do a nonlinear transformation into a way higher dimensional area and solve the matter there with the linear SVM technique. Once you have your answer, you will remodel the linear hyper plane into the x area wherever the linear hyper plane is going to be displayed as a “snake” that is separating the points. The support vectors within the higher dimension are going to be those in x area with a positive alpha.

Boser et al [101] advised to use a kernel trick to the most margin downside to form the SVM capable transformations into infinite dimensions while not having to pay the procedure price of the transformation or the price for calculate the real.

The kernel trick may be a technique wherever you employ a kernel function that represent $x^T x$ in an exceedingly non-specified higher dimensional, this while not having to pay the procedure price for the transformation.

4.3.4.1 Multiclass SVM

One of the drawbacks of the initial SVM is that it will only divide the information into two classes. Whereas it resolves this separation all right, knowledge will sometimes belong to a high variety of categories which might create the usage for this algorithmic rule terribly restricted if there was not the simplest way around this. Two of the most popular ways around this is often Winner Takes All-SVM (WTA- SVM) or Max Win Voting (MWV-SVM) [102].

WTA-SVM can construct *N* variety of classifiers wherever every classifier can check its label against all of the opposite labels at an equivalent time. The category with the upper output function is going to be the category that assigns the class.

MWV-SVM on the other hand can build one classifier for each possible attempt of signatures. The name that "wins" the preeminent orders is going to be the particular case that allots the class.

Merits: Amidst dynamic supervised learning calculations intended for TC SVM has been distinguished together of the chief compelling content order ways on the grounds that it is primed to oversee titan zones of alternatives and high generalization capability.

Demerits: However this makes SVM equation nearly extra muddled that progressively requests time and memory utilizations all around training stage and order stage.

4.4 Unsupervised Learning

Unsupervised learning refers to situations where the objective is to construct decision boundaries based on unlabeled training data to find the natural groups or clusters that exist in the data set. Unsupervised classification or clustering is a very difficult problem because data can reveal clusters with different shapes and sizes. To compound the problem further, the number of clusters in the data often depends on the resolution with which we view the data. In theoretical terms, we have a tendency to may envision of the instructor as having information of the earth, immediately information being outline by a gathering of information yield samples. The surroundings with its attributes and model are, be that as it may, obscure to the preparation framework. It works on untagged information sets to get the common groups inside the learning set. "Characteristic" is regularly sketched out explicitly or verifiably inside the bunch framework itself, and given a particular set of examples or value capacity; totally diverse group calculations cause distinctive groups. Sort choice the client can set the theorized extent of different bunches earlier time. Its sort choice suitable for planet issues in light of the fact that it is troublesome to get the named information sets.

Unsupervised learning refers to things wherever the target is to build choice limits backed untagged honing learning to inquiry out the characteristic groups or bunches that exist inside the information set [103]. Unsupervised order or bunch may be an appallingly troublesome inconvenience as an after-effect of learning will uncover bunches with totally diverse shapes and sizes. To intensify the matter more, the measure of groups inside the information sort choice relies on upon the determination with that we have a tendency to peruse the data.

In unsupervised learning a higher-request connected math model is learnt from a gathering of illustrations, with the point of demonstrating concealed reasons and density estimation. Requisitions differ from learning mental picture to information handling and information finding. These procedures are capably connected with the connected math field of group investigation, wherever through the year's sizable measure of bunch ways are anticipated. In bunch examination found out information is composed into substantive structures or scientific classifications. The target is to kind specimens into groups or groups, in place that the level of companionship is influential between parts of consistent bunch and frail between parts of different bunches. A great amount of examination has been carried out on group dissection giving a few unintentional methodologies to look for these groupings, or packs.

In this part we have a tendency to examine some basic issues and state of art connected with unsupervised learning. When talking about group investigation we have a tendency to portray the sorts of bunch calculations open inside the writing with exceptional stretch on k-methods bunch, and the thought for organizing toward oneself guide (SOM) emulated by k-methods group.

4.4.1 Cluster Analysis

Cluster examination may be an indispensable and once in a while required in planet issues. The pace, obligation, and consistency with that a bunch equation will sort out gigantic measures of data speak to overpowering motivations to utilize it as a part of provisions like information handling, insight recovery, picture division, indicator process. As a result, numerous bunch calculations are anticipated inside the writing and new group calculations still appear. The majority of those calculations are backed an) unvarying squared-blunder bunch or b) grouped positioned group. Positioned methods sort out information throughout a settled arrangement of groups which may be shown inside the style of a genogram or a tree. Squared-slip parcelled calculations organize to obtain that parcel that boosts the between-group diffuse.

It has a variety of goals. All identify with aggregating or sectioning a set of articles into subsets or bunches, specified those at interims each cluster is extra nearly connected with one another than items allocated to totally distinctive groups. Co-partner article is portrayed by a gathering of estimations, or by its significance diverse articles. Moreover, the objective is sort choice to adjust the groups into a characteristic pecking order. This includes gathering the groups themselves in place that at each level of the order, bunches at interims steady group is extra much the same as each other than those in a few groups. Bunch investigation is furthermore acclimated sort distinct detail to figure out if or not or not the data comprises of gathering different subgroups, each group speaking to protests with impressively totally diverse properties. This recent objective needs co-partner appraisal of the level of refinement between the items appointed to the unique bunches.

4.4.1.1 Cluster Distance Measures

Integral to any or the sum of the objectives of group investigation calculations is that the thought of the level of comparability (or disparity) between the distinct items being bunched. A bunch strategy tries to group the items underpinned the meaning of likeness gave to that. This will singularly return from subject material underneath issues. Throughout this area we have a tendency to examine the variability of separation measures used in bunch calculations.

4.4.1.2 Proximity Measures

Once in a while the data is outlining specifically regarding the closeness (alikeness or fondness) between sets of articles. These are either likenesses or dissimilarities (contrast or absence of natural inclination). As an illustration, in science analyses, members are asked to assess by what amount beyond any doubt articles differ from one another.

Dissimilarities will then be processed by averaging over the social affair of such judgments. This kind of data is portraying by partner *Y x Y* lattice *D*, wherever *Y* is that the reach of items and each segment d$_{ii}$ records the vicinity between the i^{th} and *i*'th protests. This framework is then given as information to the group equation. Most calculations assume a grid of dissimilarities with non- negative sections and nil slanting components: $d_{ii} = 0,\ i = 1, \ldots, Y$. On the off chance that the first information were gathered as likenesses, an adequate monotone-diminishing work is acclimated believer them to dissimilarities. Additionally, most calculations expect cruciform non likeness lattices, in this manner if the first framework *A* is not cruciform it should get traded by $(A + A^T) / 2$. Naturally judged dissimilarities are seldom removes inside the strict sense, since the Triangle distinction *dii' ≤di'k for all k € {1,..., Y}* doesnot hold. Subsequently, a few calculations that expect separations can't be utilized with such information.

4.4.1.3 Dissimilarities Supported Attributes

Frequently we've got estimations x_{ij} for examples $i = 1,\ldots, Y$, on choices $j = 1,\ldots, x$. Since the greater part of the favoured group calculations take a no similitude grid as their info, we have a tendency to first build pair savvy divergences stuck between observations. Within the most typical case, we tend to outline a non-similarity $d_j\ (x_{ij},\ x_{ij})$ between values of the j_{th} variable, and so outline

$$D(x_i, x'_i)^{\cdot} = \sum_{j=1}^{p} d_j\left(x_{ij}, x'_{ij}\right). \tag{4.27}$$

However, different decisions are attainable, and might cause doubtless completely different results. For non-quantitative attributes (e.g., categorical data), square distance might not be applicable. Additionally, it is generally fascinating to weigh attributes otherwise. Here we tend to discuss alternatives in terms of the attribute type:

Quantitative variables: Measurements of this sort of options or variables or attributes are delineate by continuous real-valued numbers. It is natural to outline the “error” between them as a monotone increasing function of their absolute distinction

$$d(x_{i'} - x_{i'}) = I(|x_{i'} - x_{i'}|)\,.$$

(4.28)

Besides square error loss $(x_i — x_{i'}))^2$, a standard alternative is that the identity (absolute error). The previous places additional stress on larger variations than smaller ones.

Ordinal variables: The values of this sort of variable are type decision delineates as contiguous integers, and therefore the realizable values square measure thought of to be associate ordered set. Examples are tutorial grades (*U, V, W, X, Y*), degree of preference (cannot stand, dislike, OK, like, terrific). Rank knowledge is a special quite ordinal knowledge. Error measures for ordinal variables are usually outlined by commutation their M original values within the prescribed order of their original values. They are then treated as quantitative variables on this scale.

Categorical variables: With unordered categorical (also decisional nominal) variables, the extent of disparity between pairs of values should be delineating explicitly.

If the variable assumes *M* distinct values, these is organized during a cruciform *M* • *M* matrix with components $L_{rr'} = L_{r'r,} L_{rr,} = 0, L_{rr'} \geq 0$. The foremost common decisions are $L_{rr'}$ = *one for all r ≠ r*, whereas unequal losses is accustomed emphasize some errors over others.

4.4.1.4 Object Dissimilarity

Next we tend to outline a procedure for combining the p-individual attribute dissimilarities $d_j(x_{ij},x_{i'j}), j = 1,... ,p$ into one overall live of no similarity $D\ (x_{i,},\ x_{i'})$ between two objects or observations $(x_i,x_{i'})$ possessing the individual attribute values. This is often nearly continually done by means that of a weighted average (convex combination)

$$D(x_{i,}x'_i) = \sum_{j=1}^{p} w_{j.}d_j(x_{ij,}x_{i'j}) , \tag{4.29}$$

where w_j may be a weight assigned to the j_{th} attribute control the relative influence of that variable in determinant the no similarity between objects.

This alternative should be supported subject material issues. If the goal is to get natural groupings within the knowledge, some attributes might exhibit additional of a grouping tendency than others. Variables that are additional relevant in separating the teams should be assigned the next influence in shaping object no similarity. Giving all attributes equal influence during this case can tend to obscure the teams to the purpose wherever a cluster formula cannot uncover them.

Although straightforward generic prescriptions for selecting the individual attribute dissimilarities d_j $(x_{ij}, x_{i'j})$ and their weights w_j is comforting, there is no substitute for careful thought within the context of every individual drawback. Specifying associate appropriate non similarity live is much additional vital in getting success with cluster than alternative of cluster formula. This facet of the matter is emphasised less within the cluster literature than the algorithms themselves, since it depends on domain data specifics and is a smaller amount amenable to general analysis. Finally, examination have missing values in one or additional of the attributes. The foremost common methodology of incorporating missing values in non-similarity calculations as in equation 4.33 is to omit every observation combine x_{ij}, $x_{i'j}$ having a minimum of one price missing, once computing the non-similarity between observations x_i and x_i'. This methodology will fail within the circumstance once each observation has not any measured values in common. During this case each observation might be deleted from the analysis. As an alternative, the missing values might be imputed using the mean or median of each attribute over the non-missing data. For categorical variables, one could consider the value missing as just another categorical value, if it were reasonable to consider two objects as being similar if they both have missing values on the same variables.

4.4.2 Clustering Algorithms

Clustering algorithms split or divide knowledge into natural teams of objects. By natural it implies that the objects during a cluster should be internally just like one another, however disagree considerably from the objects within the different clusters. Most cluster algorithms manufacture crisp partitioning, wherever every knowledge sample belongs to precisely one cluster. To mirror the inherently obscure nature of clustering, there also are some algorithms wherever every knowledge object might belong to many clusters to a variable degree. Otherwise to trot out the complexness of real knowledge sets is to construct a cluster hierarchy. Cluster might rely upon the extent of detail being ascertained, and therefore a cluster hierarchy might, a minimum of in theory, be higher at revealing the inherent structure of the information than an immediate partitioning a pair of four.

Thus, cluster algorithms are primarily divided into two types: hierarchical and partitional. Ranked cluster algorithms notice clusters one by one. The ranked ways is more divided to clustered and divisive algorithms, reminiscent of bottom-up and top-down methods. Clustered cluster algorithms merge clusters along one at a time to make a cluster tree that finally consists of one cluster, the full knowledge set. The algorithms consist

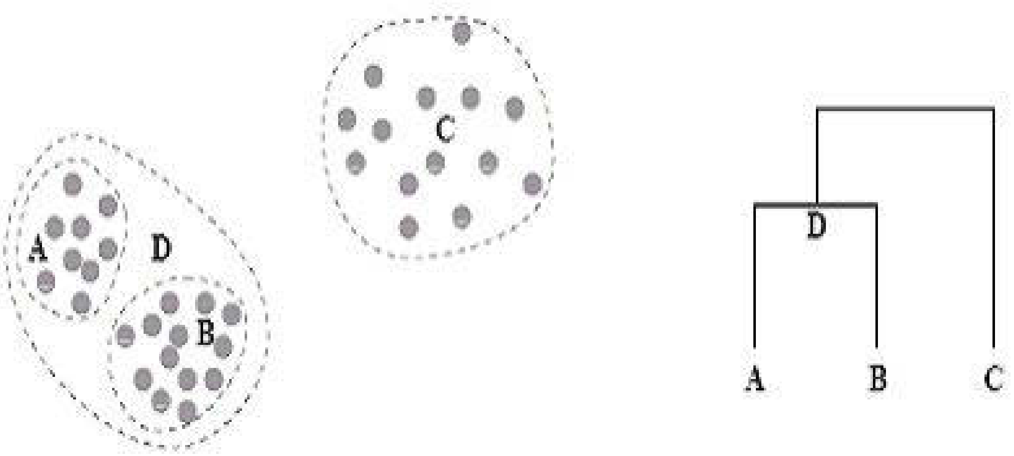


Figure 4.7 - **Fascinating clusters might exist at many levels. Additionally to A, B and C, in addition the cluster D, that may be a combination of A and B, are fascinating of the subsequent steps.**

Ranked Cluster

1. Initialize: assign every vector to its own cluster, or use some initial partitioning provided by another cluster formula
2. Figure distances *d* (C_i,C_j) between all clusters
3. Merge the 2 clusters that are flanking to every different
4. Come to step a pair of till there is only one cluster left

4.4.2.1 Partitional Cluster Techniques

Partitional Cluster algorithms turn out un-nested, non-overlapping partitions of documents that sometimes domestic decisions optimize a cluster criterion. The overall methodology is as follows: given the quantity of clusters *k*, associate initial partition is constructed; next the cluster answer is refined iteratively by moving documents from one cluster to another. Within the following subsections we tend to discuss the foremost widespread partitional algorithmic program *k*-means, and its variant bisecting k-means that has been applied to cluster documents by Steinbach et al. [104] and has been shown to usually outmatch clustered stratified algorithms.

4.4.2.1.1 K-Means Cluster

The idea behind the *k*-means algorithmic program, mentioned by Hartigan [105], is that every *k* clusters can be represented by the mean of the documents assigned to that cluster, which is called the centred of that cluster. It is mentioned by Berkhin [106] that there are units of two versions of k-means algorithmic program well-known. The primary version is that the batch version and is formally known as Forgy's algorithmic program [106]. It consists of the two major iterations:

(a) Relocate each and every one document to their nearby centred

(b) Recomputed cancroids of new assembled teams

Before the iterations begin, first *k* documents are selected as the initial centroids. Iterations continue until a stopping criterion such as no reassignments occur is achieved.

Second version of *k*-means algorithmic program has been used that is thought as on-line or progressive version. It is mentioned by Steinbach et al. [104] and Berkhin [106] that on-line k-means performs higher than the batch version within the domain of text document collections. Initially, *k* documents from the corpus are selected randomly as the initial centroids. Then, iteratively documents area unit assigned to their nearest centre of mass and centred area unit updated incrementally, i.e., when every assignment of a document to its nearest centre of mass. Iterations stop, once no reassignments of documents occur.

We outline the centre of mass vector *c* of cluster *C* of documents as follows:

$$c = \frac{\sum_{d\epsilon c} d}{|C|}, \tag{4.30}$$

consequently, *C* is obtained through averaging the weights of the terms of the documents in *C*. Analogously; we can describe the resemblance connecting a document *d* and a centred vector *c* through cosine resemblance live as

$$\cos(d, c) = \frac{dc}{\|d|C|\|}. \tag{4.31}$$

Note that though documents area unit are of unit length, centre of mass vectors are not essentially of unit length.

4.4.2.1.2 Bisecting K-Means

Even though bisecting *k*-means is truly a dissentious cluster algorithmic program that achieves a hierarchy of clusters by repeatedly applying the fundamental *k*-means algorithmic program, we tend to discuss it during this section because it could be a variant of k-means.

In every step of bisecting k-means a cluster is chosen to be split and it is split into two by applying basic *k*-means for $k = 2$. The most important cluster, that is the cluster containing the most variety of documents, or the cluster with the smallest amount overall similarity are often chosen to be split. We tend to performed experiments in each ways in which and discovered that they perform equally. So, within the experiment results section we tend to reveal solely the results of the case once the most important cluster is chosen to be split.

4.4.2.2 Hierarchical Cluster Techniques

Hierarchical cluster algorithms turn out a cluster hierarchy named a dendogram[106]. These algorithms are often categorised as dissentious top to down and agglomerative bottom-up [106]. We tend to discuss these approaches within the following subsections.

4.4.2.2.1 Divisive Hierarchical Cluster

Divisive algorithms begin with one cluster of all documents and at every iteration split the foremost acceptable cluster till a stopping criterion like a requested variety *k* of clusters is achieved.

A method to implement a dissentious stratified algorithmic program is delineating by dramatist and Rousseeuw [108]. During this technique in every step the cluster with the most important diameter is split, i.e. the cluster containing the foremost distant combine of documents. As we tend to use document similarity rather than distance as proximity live, the cluster to be split is that the one containing the smallest amount similar combine of documents. Inside this cluster the document with the smallest amount average similarity to the opposite documents is removed to create a brand new singleton cluster. The algorithmic program returns by iteratively assigning the documents within the cluster being split to the new cluster if they need larger average similarity to the documents within the new cluster.

4.4.2.2.2 Agglomerative Hierarchical Cluster

Agglomerative cluster algorithms begin with every document in a very separate cluster and at every iteration merge the foremost similar clusters till the stopping criterion is met. They are principally categorised as single-link, complete-link and average-link depending on the strategy they outline inter-cluster similarity. Figure 4.8 illustrates the idea:

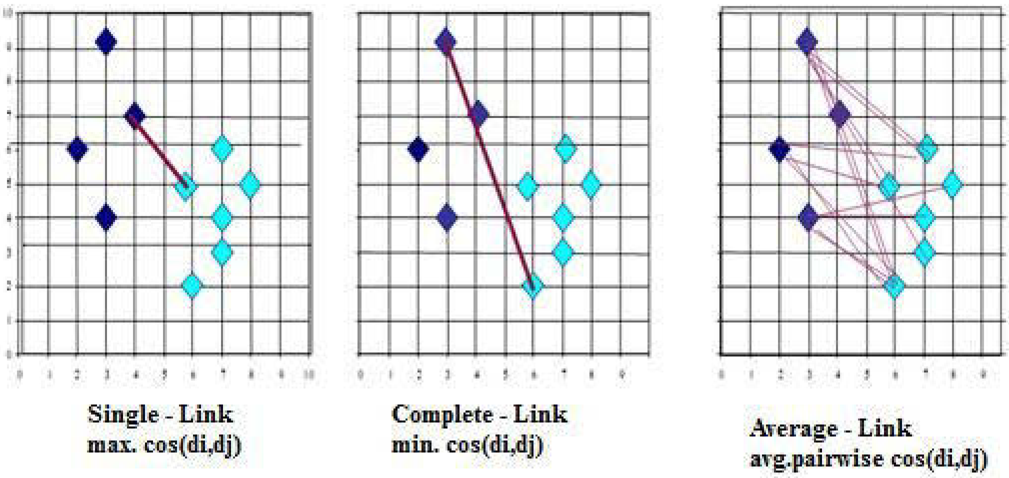


Figure 4.8 - **Inter-cluster similarity outlined by single-link, complete-link, and average-link.**

Single-Link: The single-link methodology defines the similarity of two clusters C_i and C_j because the similarity of the two most similar documents $d_i \in C_i$ and $d_j \in C_j$:

$$similarity_{single-link}(C_i, C_j) = \max_{d_i \in C_i, d_j \in C_j} |cos(d_i, d_j)| \, .$$

(4.32)

Complete-Link: The complete-link methodology defines the similarity of two clusters C_i and C_j because the similarity of the two most similar documents $d_i \in C_i$ and $d_j \in C_j$:

$$similarity_{complete-link}(C_i, C_j) = \min_{d_i \in C_i, d_j \in C_j} |cos(d_i, d_j)| \, .$$

(4.33)

Average-Link: The average-link methodology defines the similarity of two clusters C_i and C_j because the average of the pair wise similarities of the documents from every cluster:

$$similarity_{average-link}(C_i, C_j) = \frac{\sum_{d_i \in C_i, d_j \in C_j} d_i, d_j \, |cos(d_i, d_j)|}{n_i n_j}, \qquad (4.34)$$

where n_i and n_j area unit sizes of clusters C_i and C_j respectively.

4.5 Hidden Markov model

To explain the Hidden Markov Model, conjointly referred to as HMM, one has to begin to clarify the Markov process. A Markov process may be a memory less theoretical account with finite states (possible outcomes), wherever the end result of each modification activity depends only on the last state visited and wherever the possibilities does not change over time. As a simple example of the Markov process, imagine studies being created on searching habits for brand new mobile phones. Thirty five percent of who bought a brand new phone last year does not get another phone this year. Meanwhile, twenty five percent of the those who did not get a brand new phone last year can get one this year. If *40000* people buys a phone and *100,000* does not get one in exact year, how can the distribution seem like in that the year? By exploitation equation 4.35 below, it is attainable to calculate this.

z1 = people who can shopping for a brand new phone that year

z2 = those who will not get one that year

$$Cx = z$$

$$C = \begin{Bmatrix} .65 & .25 \\ .35 & .75 \end{Bmatrix}$$

$$X = \begin{Bmatrix} 40000 \\ 100000 \end{Bmatrix}$$

$$z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$$

$$\begin{bmatrix} .65 & .25 \\ .35 & .75 \end{bmatrix} * \begin{bmatrix} 40000 \\ 100000 \end{bmatrix} = \begin{bmatrix} 51000 \\ 89000 \end{bmatrix}. \tag{4.35}$$

As seen in equation 4.35, *51000* people can get a brand new phone in that year whereas *89000* will not be shopping for one.

If one wish to calculate the distribution when five, twenty or *n* years, all one got to do is replace *x* with *b* and use above equation.

The Hidden Markov Model is just like the Markov process, except that the HMM hides its current states for the observer. Associate example of the Hidden Markov Model is often seen in Figure 4.9 below. Only the output from every state is going to be visible.

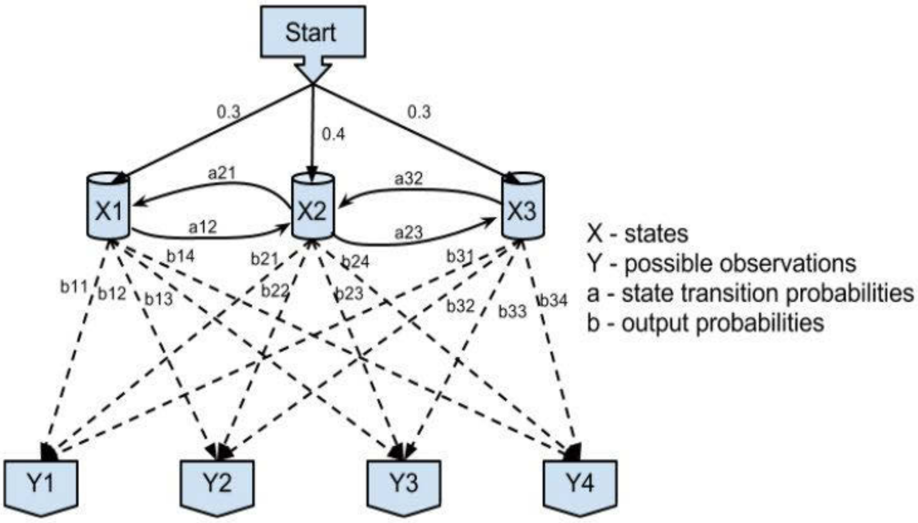


Figure 4.9 - **A example of Hidden Markov Model**

A common drawback to explain the HMM conception is that the Urn problem. During a space there is a finite range of urns states, every with an acknowledged assortment of coloured balls observations.

Someone selects an urn and haphazardly devour one amongst the coloured balls from the active urn. The colour of the chosen ball is documented on a paper and therefore the ball is swap into the urn it came from. This is often then continued to finite range of times. At any time *t*, you are allowed to appear on the paper where you will be able to see the generated output to date. the method of choosing the urn might depend upon the last choice, every attainable transition to a different urn have a set likelihood and therefore the total transition likelihood between any few urns should add to at least one.

4.6 Self-Organizing Maps

Self-Organizing Maps (SOM) is data analysis and visualization technique in machine learning proposed by Professor Kohonen (1990, 1995). A self-organizing map makes a map of one or two dimensions to lessen the estimations of high dimensional data and after that it makes the assembly of similar input together on the map. It plans the resulting clusters on a cross section. It is a combative framework where goal is to change an information data set of large size to a one or two-dimensional topological map [10]. By centred we intimate that the neurons in the SOM that are most like the information are changed as per match the incorporate more. The self-arrangement on map shows similarities of things or plans.

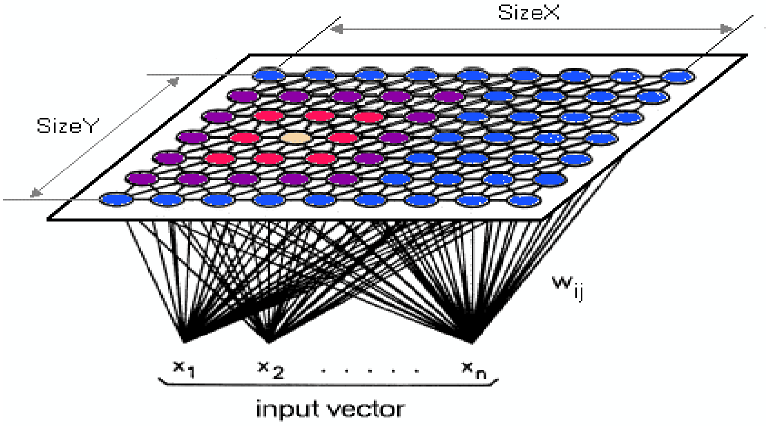


Figure 4.10 **- Structure of SOM [8].**

The structure of the SOM is a basically single support forward framework, where everyone information quality at the data layer is connected with the entire yield things at the yield layer. The yield layer is generally a two size system of yield units. The measure of sizes of information layer is higher than number of estimations of yield layered. In SOM each neuron of the yield layer is spoken to by weight vectors with estimations proportional to the measure of the information event. Each neuron is joined with neighbouring neuron with a territory connection who describes the structure of the map.

4.6.1 Learning Rule for SOM

The learning rule of SOM starts by presenting the weight vectors of SOM by periodic qualities. After presentation, the going with steps is taken to set up the SOM.

Choose a vector *x* from planning data in cyclic ask for and present to the map

Distance Measure: Calculate the partition of the data vector to the SOM to find the best matching unit. Typically used partition is Euclidean distance. It is specified as:

$$d_{ij} = \sqrt{(x_1 - w_{ij1})^2 + ... + (x_n - w_{ijn})^2} \quad , \tag{4.36}$$

where d_{ij} is the partition of data neuron *x* of n estimations to the neuron w_{ij} of the yield layer in the SOM, *i* and *j* are the coordinates of the weight vector on the map. The SOM neuron with the base partition to the data neuron is designated as winner neuron *d(k₁,k₂)*, where *k₁* and *k₂* are indices of winner neuron.

$$d(k_1,k_2)=\min_{i,j} d_{ij}\,. \tag{4.37}$$

1. **Update Rule:** Once the best matching unit has been discovered, the accompanying step is to update the victor neuron and its neighbour to be more like the data neuron. The overhaul standard is given as

$$w_{ij}(t+1)=w_{ij}(t)+\alpha(t)h(\rho,t)(X^{l}(t)-w_{ij}(t)\,, \tag{4.38}$$

where *α(t)* is a learning rate limit. It similarly decreases with time. *h(ρ,t)* is the territory limit. It is given as

$$h(\rho,t)=\exp\left(\frac{\rho^2}{\sigma^2(t)}\right)\left(1-\frac{2}{\sigma^2(t)}\rho^2\right), \tag{4.39}$$

where *ρ* is the Euclidean division of the victor neuron to the region neuron. *σ* is the compass of the range.

The fundamental SOM figuring could be shown by an essential case [1]. Ponder a fundamental 4 by 4 map. The information is one dimensional. An illumination x = 6.4 is send to the map. Here we pick *σ = 1, α(t) = 0.5.* In the wake of processing the Euclidean partition of information with each neuron of the SOM, we find that weight vector with value six is the winner neuron. Immediately the victor neuron nearby its 4 neighbours is overhauled using the update guideline. The figure underneath graphically demonstrates the overhauling strategy,

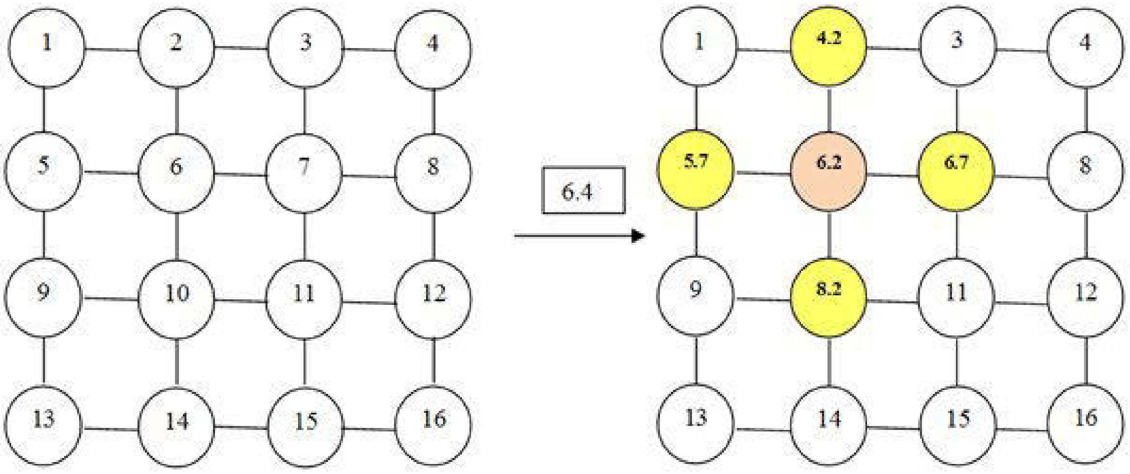


Figure 4.11 **- SOM Learning Example**

The Self Organizing Map is a champion around the most renowned neural framework frameworks and it is comprehensively used inside mixture of demand. These orders may incorporate examination of data. Various business framework intrusion recognizable proof procurements also use SOM estimation to recognize strike affiliations in the framework packages. SOM is also used as a pre-processor for unsupervised learning calculation [9].

Self-Organising Map (SOM) could be a neural network model that supports unsupervised learning. It is an efficient technique for clustering and classifying high dimensional knowledge. It well-tried to be a valuable tool in data processing and information discovery in Databases (KDD) [108]. SOM training algorithm has applications in pattern detection, image analysis, method watching, organization of document collections etc. In variety of information analysis cases associated with social science are given within which the SOM training algorithm has been a crucial tool. Additional samples of fruitful usage of the SOM training algorithm in numerous engineering tasks are found as an example. A comprehensive listing of SOM training algorithm analysis has been compiled by Kaski et al. [110].

A SOM training algorithm consists of neurons organized on a low-dimensional grid. The amount of neurons will vary from a number of dozen up to many thousand. Every vegetative cell is portrayed by a d-dimensional weight vector additionally referred to as paradigm vector or codebook vector, wherever d is adequate the dimension of the information vectors. The neurons are associated with adjoining neurons by a territory connection that directs the topology, or structure, of the guide. The topology is comprehensively talking isolated to two components: local grid structure and the worldwide guide structure [111]. Examples of rectangular and hexangular cross section structures are demonstrated in figure 4.12, and specimens of completely distinctive mixtures of guide shapes in figure 4.13.

The SOM preparing calculation honing equation looks like the vector quantisation (VQ) calculations, in the same way as k-methods. The fundamental refinement is that furthermore to the best- matching weight vector is extended towards the given training specimen, as in figure 4.13. The tip comes about that the neurons on the network get requested: neighbouring neurons have comparable weight vectors.

Since the trouble vectors of the SOM preparing calculation have exact squat dimensional organizes on the guide network, the SOM preparing calculation is furthermore a vector projection technique. Along the ideal model vectors and their projection diagram an espresso dimensional guide of the illumination complex.

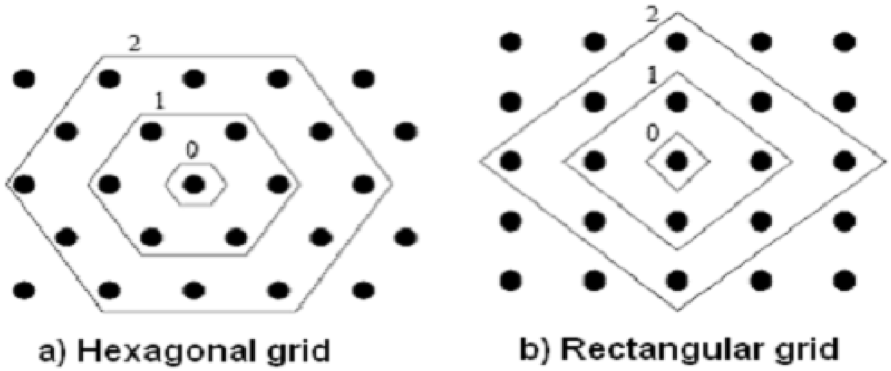


Figure 4.12 **- Separate neighbourhoods (size O,1 and 2) of the axis most unit: (a) hexangular lattice, (b) rectangular lattice. The intimate polygon figure corresponds to 0-neighbourhood, the instant to the 1-neighbourhood and also the largest to the 2- neighbourhood**

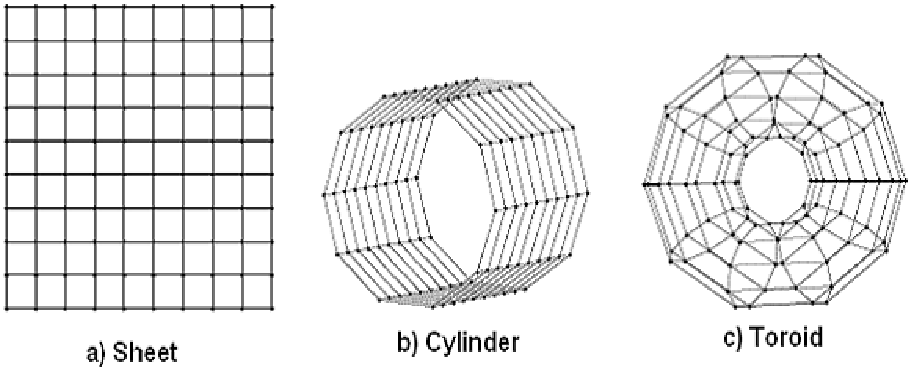


Figure 4.13 - **Totally different map shapes. The default forms (a) and two shapes anywhere the map topology accommodates spherical data: cylinder (b) and toroid (c).**

4.6.2 SOM Algorithm

The training algorithm is simple, solid to missing qualities, and - perhaps generally imperatively. It is easy to check the guide. These properties make SOM a recognized apparatus in information transforming, learning investigation and group mental picture segment. The logic of the SOM calculation is as takes after:

a) Initial Stage: Foremost of all, it initialises all the neurons within the map with the data vectors aimlessly.

b) Data Normalization: For a stronger distinguishing proof of the groups the insight need to be standardized. We tend to utilized the "extent" procedure wherever all aspects of the illumination vector is standardized to abide the in travel [*0,1*].

c) SOM Training: Choose information vector *x* from the data set aimlessly. A best matching unit (BMU) for this data vector, is found inside the guide by the consequent metric.

$$\|x - m_c\| = \min_i\{\|x - m_c\|\}\,, \tag{4.40}$$

where m_c is the reference vector identified with the unit *i*.

d) Change Step: The reference vectors of BMU and its neighbourhood are overhauled in keeping with the consequent tenet

$$m_i(t+1) = \left\{ \begin{matrix} m_i(t) + \alpha(t).h_{ci}(t).[x(t) - m_i(t)], \begin{matrix} i \in N_c(t) \\ i \in N_c(t) \end{matrix} \\ m_i(t) \end{matrix} \right\}, \tag{4.41}$$

where $h_{ci}(t)$ is the part neighbourhood function of the winner neuron c and increases with time *t*.

x(t) is an information vector randomly drawn from the data workstation record set at time *t*. *α(t)* is that the learning rate at time *t.*

$N_c(t)$ is that the area situated for the victor unit *c*.

The surpassing mathematical statement details BMU and its neighbourhood move closer to the information vector. This change in accordance with information vector structures the reason for the bunch establishment inside the guide.

e) Data Squad Visualisation: Steps three and four are enduring for chose extent of trials or ages. At the point when the ways are finished the guide unfolds itself to the dissemination of the information set discovering the measure of common groups exists inside the information set. The yield of the SOM is that the situated of reference vectors identified with the guide units. This set is termed as a codebook. To take a gander at the groups and likewise the outliers ran across by the SOM we have to check the codebook. U-Matrix is that the procedure generally utilized for this reason.

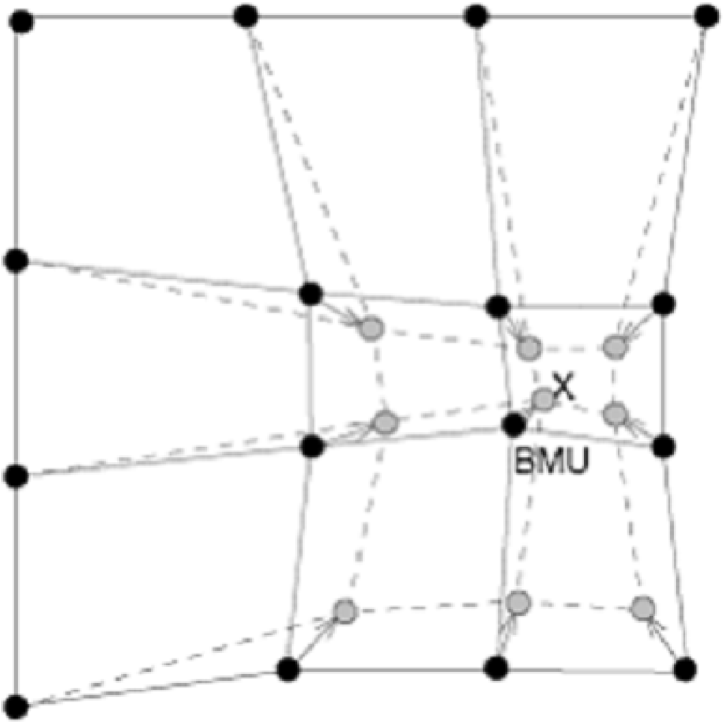


Figure 4.14 **- Change the least difficult matching unit (BMU) and its neighbours towards the data test with x. The strong and dabbed lines compare to situation before and when change, respectively.**

• **Missing Values:** Within the proposition, this is depicted by the worth of NAN inside the vector or information grid. Missing components are taken care of by simply barring them from the crevice figuring. It is expected that their commitment to the crevice $||x$ — $mi||$ is *0*. In keeping with, this is regularly a true blue determination in light of the fact that the same variables are unnoticed in every separation count over that the base is taken.

• **Mask:** Every variable has partnered weight subject. This is regularly principally utilized in double kind for avoiding beyond any doubt variables from the BMU-discovering technique (1 for exemplify, zero for prohibit). On the other hand, the cover will get any qualities; hence it is utilized for weight variables within keeping with their significance. With these progressions, the whole live gets to be:

$$\|x - m_i\| = \sum_{k \in K} w_k (x_k - m_k)^2 \, , \tag{4.42}$$

where *K* is that the situated of radiant variables of specimen vector *x*, *xk* and *mk* are the k^{th} components of the example and weight vectors and w_k is that the k_{th} veil worth (mask(k)). At the point when discovering the BMU, the weight vectors of the SOM calculation is redesigned so the BMU is blended closer to the information vector inside the data house. The topological neighbours of the BMU are dealt with similarly. This adjustment methods extends the BMU and

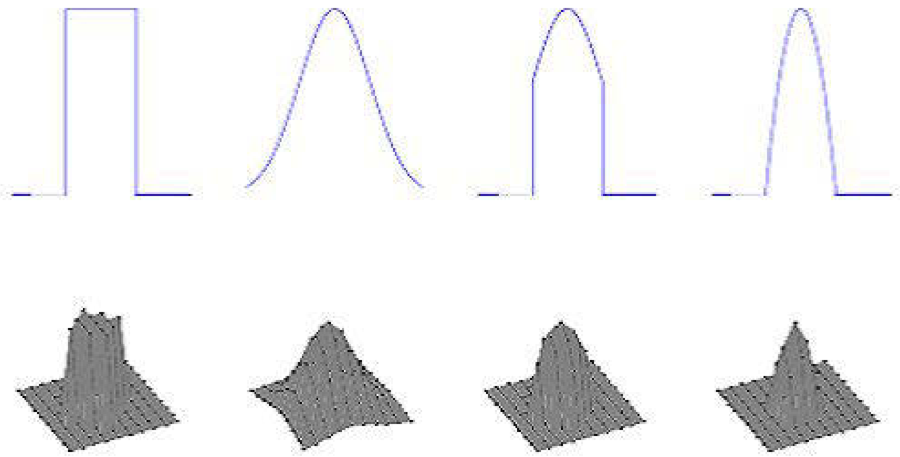


Figure 4.15 **- Completely diverse neighbourhood capacities.**

Different neighbourhood functions. From the left 'bubble' h_{ci} (t) = 1 (<r_t — d_{ci}), 'gaussian' $\boldsymbol{h_{ci}}$ (t) = $\boldsymbol{e^{-d}d^{/2a}t}$, 'cut-gauss' $\boldsymbol{h_{ci}}$ (t) = $\boldsymbol{e^{-d}d^{/2a}t}$ 1 $\boldsymbol{(a_t}$ — d_{ci}), and 'ep' h_{ci} (t) = maxj0,1 — (a_t — $d_{ci})^2$ j, d_{ci} = ||r_c — rj|| is the distance between map units c and i on the map grid and 1 (x) is the step function: 1 (x) = 0 if x < 0 and 1 (x) = 1 if x > 0. The top row shows the function in 1-dimensional and the bottom row on a 2-dimensional map grid. The neighbourhood radius used is <r_t = 2.

The area span utilized is a couple. Its topological neighbours toward the outline vector as uncovered in figure 4.16. As listed in the SOM algorithm, the SOM update rule for the weight vector of unit *i* is:

$$m_i(t+1) = m_i(t) + \alpha(t)h_{ci}(t)[x(t) - m_i(t)]\,. \tag{4.43}$$

The input vector *x(t)* is commitment vector self-assertively drawn from the data document set at time *t,* $h_{ci}(t)$ the area part adjust the winner neuron *c*. The area part could be a non-expanding work of your time and of the crevice of neuron *i* from the victor *c*. It orders the region of impact that the data outline has on the SOM. Completely diverse bit neighbourhoods which will be utilized with SOM are compressed as a part of Figure 4.16, *α(t)* speaks to the enlightening rate, which is also diminishing work of your time used to unite the guide to the groups uncovered by SOM inside the data set. Figure 4.16 shows the different learning rates and their conveyances. For faster learning, SOM will be prepared in an exceedingly group mode. Cluster instructing algorithmic system is moreover redundant, however instead of utilizing a solitary data vector at once, the full data set is presented to the guide before any progressions are made thereupon the name "group". In every honing step, the data set is divided off in keeping with the Voronoi areas of the guide weight vectors, i.e. each data vector has a place with the data set of the guide unit to that its adjacent. Ensuing to this, the most recent weight vectors are ascertained as:

$$m_i(t+1) = \frac{\sum_{j=1}^{n} h_{ic}(t)x_j}{\sum_{j=1}^{n} h_{ic}(t)}.$$

(4.44)

4.6.3 SOM as Clump Technique

SOM groups the data set upheld its division ability on the given commitment case set. Dissemination consolidates the preparatory data set down to an unobtrusive representative *set "straight" (robust line) α(t) = α0 (1 — t/T), "force" (spot dashed) α (t) = α0 (0.005/a0)t/T and "inv" (dashed) α (t) = α0/ (1 + 100t/T),* wherever *T* is that the training length and α^0 is that the introductory learning rate of models to figure with. The representative set of models will be used in computationally focused undertakings, in the same way as bunch or projection; to urge harsh outcomes with decreased procedure esteem [112] this diminishment is extremely paramount especially in data investigation.

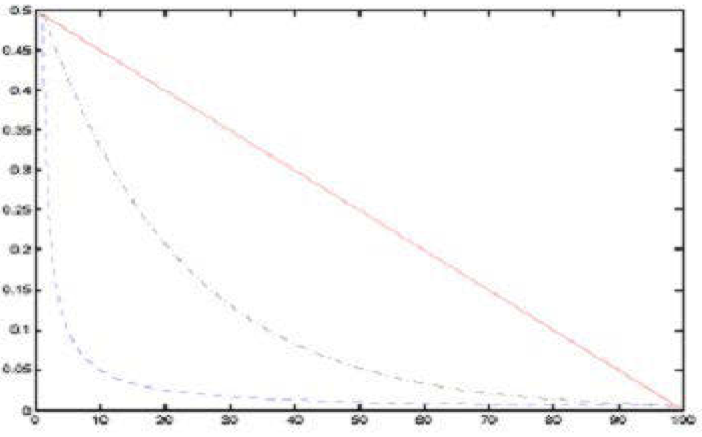


Figure 4.16 **- Dissimilar knowledge rate capacities.**

Also, since the models are moulded as midpoints of the in arrangement representations, the impact of 0-mean commotion peaceful like outliers are consolidated.

Vector division calculations, attempt and understand a set of picture vectors $m_i = 1, \ldots, m$ that recreate the introductory data set and in addition possible. The best better-known calculation to pursuit out these models is the k-implies calculation [113]. A division calculation discovers a gathering of $M = k$ picture vectors that minimize the quantisation mistake nuclear weight, will not to live the quantization property of the algorithmic system.

$$E_q = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} |x_{ij} - m_{bij}|^r \, , \tag{4.45}$$

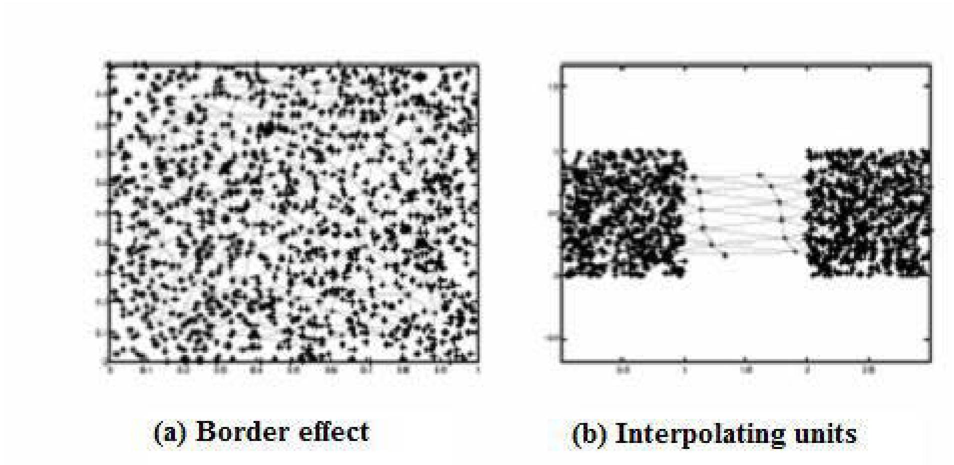
where b_i is the file of the best-matching picture, r is the separation standard.

The point density of the prototypes follows the density of the training data. Asymptotically it holds that:

$$p(m) \propto p(x)^{\frac{d}{d+r}} \, , \tag{4.46}$$

where d is that the extent and $p(x)$ *and* $p(m)$ are the prospect reduction capacities of the information document and likewise the model vectors respectively.

The SOM is nearly connected with the k-implies algorithmic system. On the off chance that the area portion value is one for the BMU and zero somewhere else $(h_{bij} = \delta\ (b_{i,j}))$. The SOM lessen to the versatile k-implies algorithmic system. Conjointly cluster guide diminishes to bunch k-implies. The refinement between established vector division and SOM is that the SOM performs local smoothing inside the area of each guide unit. This smoothing makes the requesting of the models, however once the area span is diminished.



(a) Border effect (b) Interpolating units

Figure 4.17 - **Two feature effects created by the area work: (an) outskirt impact and (b) adding units.**

The + are the guiding data, and likewise the associated framework of loops is that the guide. Throughout the instructing, it conjointly actualizes a mimicked treating style of teach subject that transforms the division strategy extra solid.

Additionally, there are two effects of Figure 4.17:

• **Border effect:** The area definition is not cruciform on the outskirts of the guide. Thus, the density estimation is completely distinctive for the fringe units than for the centre units of the guide. In watch, the guide is contracted on the fringes. This has the effect that the tails of the negligible appropriations of variables are less generally given than their focuses. In a few cases, this could encourage curtailing the effect of outliers, however typically; this might be a shortcoming of the SOM.

• **Interpolating units:** Once the data cloud is spasmodic, introducing units are the data circulation. Notwithstanding, simply if there should be an occurrence of a few dissection instruments, case in point single linkage cluster, these may offer false signals of the manifestation of the data complex and should must be forced to be deemphasized or totally not noted of study. In the event that the information data set considered about in light of the fact that the set of irregular information variables *x* that is circulated in keeping with a probability density work *p(x)* then the SOM structures co-partner degree rough guess to the current probability density work *p(x),* utilizing a limited reach of middle of mass vectors m_c *(c= 1, 2,..., k).* At present these reason densities of the models take after harshly the probability density of the data. When the middle of mass is picked, the rough guess of *x* methods discovering the inside of mass vector m_c gets closest to x vector in the data space. The perfect consequence of m_c minimizes the typical needed worth E_{avg} of the quantization slip little weight, unique as

$$E_{avg} = \int \|x - m_c\|'^2 p(x)dx\,. \tag{4.47}$$

Steady with [84], if the measure of middle of mass vectors is titan, the best decision of m$_c$ qualities is specified their motivation density approximates to

$$p(m)\alpha\, p\,(x)^{\frac{2}{3}-\frac{1}{3\sigma^2+3(\sigma+1)^2}}\,. \tag{4.48}$$

Variables assume a fundamental part in division properties of the SOM. The significance is of variables characterizes the point of view of the division. When division joins a focal part in data examination, it is crucial to comprehend what this perspective is; as a consequence of any investigation underpinned the division can repeat however well the variables are envisioned. By including, uprooting, or rescaling variables, an exceptional division result is non heritable as a consequence of the division lapse work nuclear weight changes correspondingly. However well every variable is envisioned inside the division relies on upon however capably the variable impacts the full division blunder. The division mistake nuclear weight will be communicated as far as variable-wise slips E$_j$:

$$E_q = \sum_{j=1}^{d} \frac{1}{N} \sum_{i=1}^{d} |x_{ij} - m_{bij}|^2 = \frac{1}{N} \sum_{j=1}^{d} E_j\,. \tag{4.49}$$

to live the coarseness of the division with connection to each variable, the mistakes E_j will be contrasted with divisions performed on every variable exclusively with expanding extent of quantization focuses $E_j(k)$, $k = 1,\ldots, N$. Depending on the conveyance qualities of the variable, the division failure diminishes at completely diverse rates. Case in point, for a consistently conveyed variable, the division failure diminishes in keeping with equation $E_j(k) = \alpha^2{}_j k^2$, wherever α_j is that the fluctuation of the variable. We have a tendency to utilize this SOM bunch proficiency to discover the groups of gamma radiation blasts data sets inside the research endeavour of dissection spoke to throughout this part.

4.6.4 SOM as Image Technique

One of the preeminent basic properties of the SOM is that its cohort degree conservative procedure for picture of high-dimensional data. The SOM is consequently a magnificent instrument in preparatory data investigation [113]. Graphic ways attempt and acknowledge low-dimensional arranges that safeguard the separations (or the request of separations) between the initially high-dimensional articles. A traditional projection method is multi-dimensional scaling (MDS) that tries to protect pair clever separations between all items inasmuch as decreasing the estimation. The incorrectness moves to be diminish is:

$$E_{mds} = \sum_{i=1}^{N}\sum_{j=1}^{N}(d_{ij} - d'_{ij})^2 \, , \tag{4.50}$$

where d_{ij} is that the separation between data tests i and j inside the information range $||xi - xj||$, and $d_{'ij}$ is that the comparing separation between the projections facilitates inside the yield region. There is conjointly a non-metric adaptation of MDS that tries to protect the requesting of the separations. Distinctive remarkable projection strategies are sammons' mapping, Curvilinear Component Analysis (CCA).

$$sammons'mapping\colon E_{sam} = \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{(d_{ij} - d'_{ij})^2}{d_{ij}} \, . \tag{4.51}$$

$$CCA : E_{cca} = \sum_{i=1}^{N}\sum_{j=1}^{N}(d_{ij} - d'_{ij})^2 \, e^{-dij} . \tag{4.52}$$

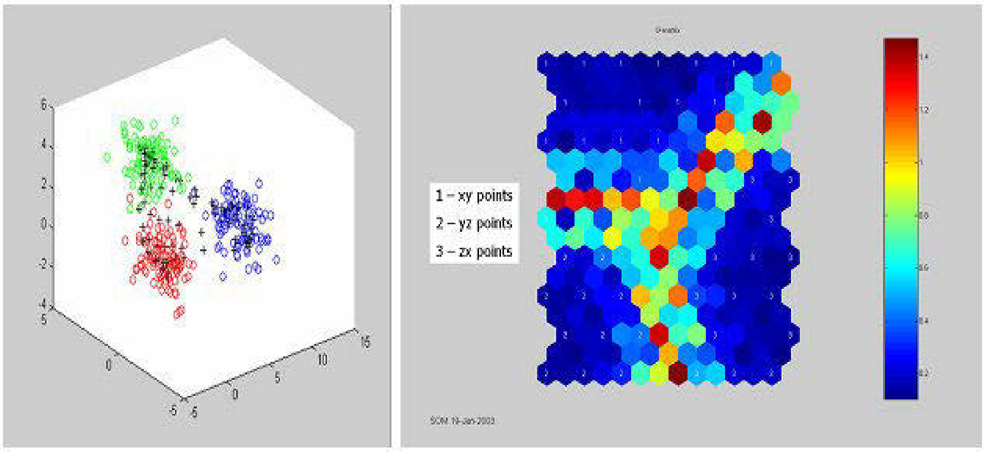
For SOM the failure work venture down is given by:

$$E_{som} = \sum_{i=1}^{N}\sum_{j=1}^{N} h\,(d'_{ij})(d^{2}{}_{ij}), \tag{4.53}$$

where h is that the area portion work. Since its monotonically diminishing work of d_j, modest separations inside the area are pushed; conjointly d_{ij} relies on upon the density conveyance of the information document set.

Along these lines, the meaning of neighbourhood in SOM tunes to include record density. Rather than attempt and protect the beginning separations, the SOM requests picture vectors on a predefined guide lattice specified local neighbourhood sets inside the projection are saved.

SOM is especially sensible at looking after the trustiness of the projection: if two data examples are close to each one in turn inside the picture, they are extra surely to be passing on the starting high-dimensional zone still. The picture procedures that are basically exploited in this thesis are represented below.



(a) Points in threed - space (b) SOM grouping with U-matrix

Figure 4.18 **- SOM Classification Points in Thread Space**

4.6.5 U-matrix

U-matrix (unified distance matrix) illustration of the Self-Organizing Map visualizes the distances between the map units or neurons. Associate degree U-Matrix displays the native distance structure of the information set. It is a typical tool for the show of the gap structures of the input file on SOM [114]. The gap between the adjacent neurons is calculated and conferred with totally different colourings between the adjacent nodes. In U-matrix colour committal to writing scale is employed to tell apart varied clusters. The clusters and their outliers, boundaries are pictured by totally different colours. This will be a useful presentation once one tries to search out clusters within the input file while not having any a priori data concerning the clusters. Teaching a SOM and representing it with the U-matrix offers a quick thanks to get insight of the information distribution while not human intervention.

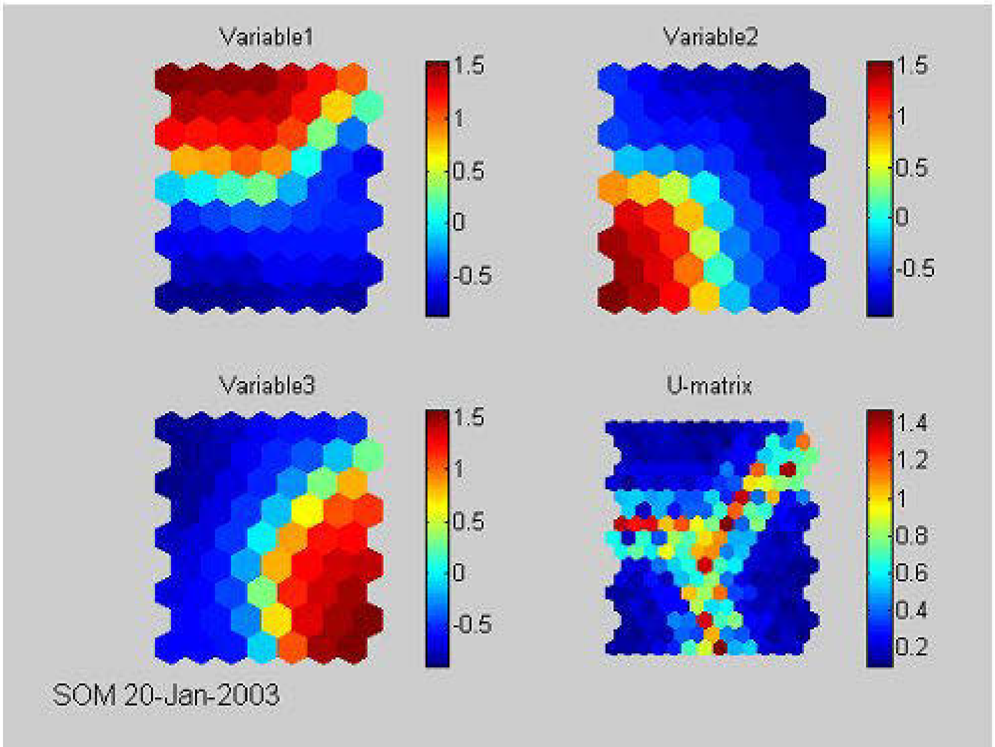


***Figure 4.19* - Component Planes for the Data Points in 3-D space.**

The U-Matrix is made on high of the map. The colour committal to writing of a map unit is predicated on the issue “U-height”. The unified distance matrix (U-matrix) visualizes all distances between every map unit and its neighbours. This can be doable attributable to the regular structure of the map grid: it is simple to position one visual marker between a map unit and every of its neighbours. The map prototypes follow the likelihood density functions of the information, the “u-height” distances are reciprocally proportional to the density of the information. Thus, cluster borders will be known as totally different colours separating the map units of low distances that are with within the cluster. This interpretation also can be utilized in clump. Let *i* be a unit on the map, NN(*i*) be the set of immediate neighbours on the map, m(*i*) the load vector related to vegetative cell *i*, then

$$U - height(i) = \sum_{j \in NN(i)} d\big(m(i) - m(j)\big), \tag{4.55}$$

where *d(m(i) — m(j))* is that the distance utilized in the SOM algorithmic program to construct the map.

Therefore U-Matrix could be a show of the U-heights on high of the grid positions of the neurons on the map. Once the u-heights are determined distinction colour committal to writing are assigned to various clusters within the following way:

-U-height(i) = mean(U — heights) => (clustercenter)
-U-height(i) < mean(U — heights) => (intercluster)
-U-height(i) > mean(U — heights) => (intracluster)
-U-height(i) < min(U — heights) => U-height(i) = zero (boundary)

Properties of the U-Matrix:

- The pose of the projections of the input file points replicate the topology of the input area, this can be transmissible from the underlying SOM algorithmic program.
- Weight vectors of neurons with giant U-heights are terribly distant from different vectors within the information area.
- Weight vectors of neurons with tiny *U*-heights are encircled by different vectors within the information area.
- The U-Matrix become conscious the materialization of structural preferences of the distances among the information area.
- Outliers, still as doable cluster structures will be recognized for top dimensional information areas.
- The correct setting and functioning of the SOM algorithmic program on the input file also can be visually checked.

Figure 4.19 shows the SOM classification of points distributed in thread area. The information set is made from the random vectors taken from a cube in 3D area. The image vectors pictured by '+' are chosen haphazardly to assign to the SOM. SOM discovered the three teams (*XY, YZ, ZX* plane points), the U-matrix shows these teams in blue colour well separated by the boundaries 4.19 (b).

4.6.6 Component Plane Visualization

Component plane illustration displays the values of every model vector component, i.e. values of every variable, on the map grid. In figure 4.19, the three element planes *(XY, YZ, ZX)* are shown. For every envisioned variable, or vector element, one SOM grid is envisioned specified the colours (or for instance sizes) of the map unit markers modification in keeping with the envisioned values. Relationships between variables will be seen as similar patterns in identical places on the element planes: whenever the values of one variable modification, the opposite variable changes, too. Though any reasonably projection can be used to link the element planes along, the SOM grid works significantly well during this task. Due to the dynamic focus of the map, the behaviour of the information will be seen regardless of the native scale. By inspecting all the element planes at the same time, one would observe relationships between variables and even roughly distinguish structure of the input file. Ordering of the element planes makes it easier to analyse an outsized range of element planes at the same time. The essential plan is to rearrange the element planes in such the simplest way that similar planes (that is, interconnected variables) lie near every other; this organization is also carried out using the SOM algorithm. Colouring of the SOM was used with the distinction that the changes within the colours between neighbouring units were chosen to replicate cluster structure of the model vectors of the map.

4.7 Summary

Below is the summary of the algorithms which are discussed above.

Naïve Bayes Classifier

The Naïve Bayes classifier may be a light-weight, quick and climbable algorithmic program that perform surprisingly well compared to the advanced models because the information set does not grow too huge. A disadvantage is that the algorithmic program can run into a retardant if you encounter information with a variable having zero chance since it will ruin your equation once increased with the opposite variables. This could but be fastened if you sleek the information beforehand by eliminating those values.

Decision Rules

Decision rule is amongst the foremost light-weight of the classification algorithms which will perform unusually well given tiny specific issues. However, it is not appropriate for classification issues wherever you have got a high variety of various attributes and values since it depends on one rule. For text classification, there are often many thousands of various words that must be classified and thence the inferring rudimentary rules classifier can presumably forever perform worse than classifying out of the blue. Yet, this algorithm may be an example that even advanced issues will have an easy answer if you are allowed to introduce error.

Decision Trees

This is a fairly clear-cut thanks to construct a classifier since it initialize the tree with one rule, and as you progress down within the tree by perpetually creating choices what branch to continue downward in, you will eventually reach the tip node being a leaf wherever the ultimate classification declare your instance lies. If this answer is enforced in an algorithmic program that may optimize the foundations at every branch by removing methods that hold none or little or no valuable data and thence maximising the amount of instances you will classify within the final leaf while not over fitting, then you will get pretty sensible generalized results.

Hidden Markov Model

You principally grapple the primary input tranquil as what goes out at time *t*, however what truly happened within may be a little bit of a mystery. Not knowing what goes on within the model will for a few applications be an enormous disadvantage.

K-means Clustering

An interesting bunch algorithmic program with a clear-cut implementation that may even be used for classification as long because the k-number of categories is that the same because the variety of categories existing within the information set.

K-Nearest Neighbour

This algorithmic program is analogous to k-means bunch since it classifies values supported the classification of the bulk of the pre-labelled information points within the nearby cluster with size *k*. It is sensible performance and because of its lazy nature it does not need any time to come up with a classification information model. The disadvantage of this can be the memory still as time quality throughout the classification method quickly will increase if you are employing a huge training-data set.

Self-Organizing Map

Self-Organising Map is a neural network model that is focused around unsupervised learning. It is an effective strategy for clustering and visualization of high dimensional statistics. It turned out to be an important instrument in data mining and Knowledge Discovery in Databases (KDD). SOM has applications in pattern distinguishment, image examination, methodology checking and so forth.

In this examination the Self Organization Map is utilized to discover attackers. The 41 features from KDD99 and from NSL-KDD datasets are utilized as input data, SOM changes 41-dimensional data information vector into 2 yields vector (0 if entrance pattern is not an assault (Normal), and 1 values for attackers (abnormal). The SOM forms those offered information to perceive kind of attacks or typical transactions.

Support Vector Machines

A fairly previous plan for a classification algorithmic program that was brought back in a very Holy Writ throughout the 90 is with very good performance. It uses an artless thanks to separate information points into two categories mistreatment vectors. this could even be applied on classifications wherever you have got quite two categories by running the algorithmic program many times, testing all the one v_s one class-combinations and so finally classifying the information mistreatment the support vector machine classifier that had the very best performance.

Having discussed the above machine learning techniques, this thesis will be using the Naïve Bayes technique under supervised learning and self-organising map under the unsupervised learning in order to detect network intrusion.

To overcome low detection rate and high false caution issues in right now existing IDS, these two methodologies are utilized to boost the execution of intrusion detection for rare and complicated attacks.

Most of the systems use Self-Organizing Maps (SOMs), while a few utilized different types of unsupervised neural nets. SOM are more powerful than static systems because, dynamic systems have memory and they might be prepared to learn consecutive different patterns. It is additionally demonstrated that utilizing SOM acquires execution within correlation with other state-of-the-symbolization detection systems.

Contrasted with the other approaches, Naïve Bayes approach attain higher detection rate, less time intensive and has ease variable. In any case, it creates to some degree all the more false positives. As a Bayesian system is a confined system that has just two layers and accept complete freedom between the data hubs. This postures an impediment to this research work.

CHAPTER 5
KDD CUP'1999 DATASET

This chapter discusses the KDD Cup 1999 dataset that has been selected for training and testing in the Network Intrusion detection systems (NIDS) for this thesis. All the attributes of dataset, conversion and manipulation techniques applied to dataset have also been discussed in this chapter.

5.1 Introduction

As Internet continues developing with an exponential pace, so network security is becoming more and more challenging. A few defensive measures, for example, firewall has been set up to check the exercises of interlopers which could not ensure the full security of the framework. Hence, there is a requirement for more dynamic component like Intrusion Detection Systems (IDS) as a second line of guard. Intrusion Detection is the procedure of checking occasions happening in a PC framework or system and investigating them for indications of intrusions. Intrusion Detection Systems are basically considered host-based or network based. The previous works on data gathered from inside a singular PC framework and the recent gather crude systems bundles as the information source from the system and investigate for indications of intrusions.

The efficiency of an Intrusion Detection System is restrained using its probability of giving a signal upon an intrusion i.e. attack detection rate and the ratio of false alarms in them. On the other hand, network forensics is about offline investigation on taken data to ascertain the source of security attacks. For this kind of offline and online traffic analysis, the detection of an attack and its correct categorization is crucial.

With the colossal development of workstation systems use and the gigantic build in the amount of provisions running on top of it, system security is getting to be progressively more significant. As it is demonstrated in, all the machine frameworks experience the ill effects of security vulnerabilities which are both in fact troublesome and monetarily immoderate to be understood by the makers. In this manner, the part of Intrusion Detection Systems (IDSs), as uncommon reason units to discover peculiarities and assaults in the system, is getting to be more imperative. The exploration in the intrusion detection field has been basically concentrated on anomaly based and misuse based detection procedures for quite a while. While misuse based detection is for the most part supported in business items because of its unoriginality and high precision, in scholastic research anomaly detection is commonly considered as the more effective technique because of its hypothetical potential for detecting to new attacks. Encoding to distinguish network intrusions secures a machine system from unapproved clients, including maybe insiders. The intrusion detection learning assignment is to manufacture a prescient model (i.e. a classifier) fit for recognizing "attack" connections, called intrusions or ambushes, and "normal" ordinary connections.

The 1998 DARPA Intrusion Detection Evaluation Program was started and oversaw by MIT Lincoln Labs. The target was to review and evaluate activities in intrusion detection. A standard set of information to be examined, which incorporates a wide mixture of intrusions mimicked in a military the earth, was given. The 1999 KDD intrusion detection challenge utilizes a rendition of this dataset [8].

Lincoln Labs set up an environment to gain nine weeks of crude TCP dump information for a neighbourhood (LAN) reproducing an average U.S. Air Force LAN. The LAN was focused like a real environment and blasted with multiple attacks.

A connection is a sequence of TCP packets starting and ending at some time duration between which data flows to and from a source IP address to a target IP address under some well-defined protocol. Also, each connection is labelled as either normal or as an attack with exactly one specific attack type. Each connection record consists of about 100 bytes. The raw training data is about four gigabytes of compressed binary TCP dump data obtained from seven weeks of network traffic. Finally, the completed process generated around five million connection records. Likewise, the two weeks of test data gives around two million connection records. For each TCP/IP connection, 41 various quantitative and qualitative features are obtained with normal and attack data.

The Association for Computing Machinery (ACM) has a specific interest group on Knowledge Discovery and Data mining (KDD) which is the most mainstream expert connection of information diggers. The KDD sorted out the yearly Data Mining and Knowledge Discovery competition called KDD Cup in diverse regions [9]. The various focused areas of KDD and its period have been tabulated in Table 5.1.

***Table 5.1* - KDD-CUP Centre of Attention**

Year	*Focused Area*
KDD-CUP 1997	Immediate showcasing for lift curve improvement
KDD-CUP 1998	Immediate showcasing revenue driven advancement
KDD-CUP 1999	Workstation system network intrusion detection
KDD-CUP 2000	Online retailer site click stream investigation
KDD-CUP 2001	Sub-atomic bioactivity and Protein district expectation
KDD-CUP 2002	Bio Medical report and Gene part characterization
KDD-CUP 2003	System mining and utilization log investigation
KDD-CUP 2004	Molecule material science; in addition to Protein homology
KDD-CUP 2005	Internet user search query categorization
KDD-CUP 2006	Pulmonary embolisms detection from image data
KDD-CUP 2007	Consumer proposals
KDD-CUP 2008	Breast cancer
KDD-CUP 2009	Quick scoring on an expansive database

The KDD Cup 1999 dataset is predicated upon the version of a dataset that was utilized by 1998 DARPA Intrusion Detection Evaluation Program. This proposed exploration work utilizes the benchmark dataset ordered for the 1999 KDD intrusion detection challenge, by MIT Lincoln Labs [115]. The primary point of interest of utilizing this dataset is that the proposed examination work is equipped for furnishing huge information that are effortlessly imparted to different specialists and engineers. The reaction got from different analysts permit to enhance the consequence of this proposed work.

In 1999, the KDD acknowledged and approved authority information because the standard benchmark information base for IDS referred to as KDD Cup99. Each connection of packet is represented by 41 features.

5.2 KDD Cup99 Methodology

As demonstrated earlier, the essential goal of this research work is to focus on the interpolation of the 41 characteristics in KDD’99 intrusion detection datasets to ambush identification (or separation of normal connection from attacks). To achieve this interpolation, a methodology dependent upon data increase is utilized. Taking into account the entropy of a characteristic, data addition measures the pertinence of a given characteristic, as it were its part in deciding the class name. Assuming that the characteristic is applicable, as such exceptionally advantageous for a faultless determination, ascertained entropies will be near 0 and the data increase will be near 1. Since data addition is computed for discrete characteristics, nonstop characteristics are discredited with the accentuation of giving sufficient discrete qualities for recognition.

5.3 NSL-KDD Data Set

The NSL-KDD data set proposed to resolve the innate issues of the KDDCUP'99 data set. KDDCUP'99 is the frequently broadly used data set for anomaly detection. Yet Tavallaee et al directed a measurable dissection on this data set and discovered two vital issues that significantly influenced the execution of assessed frameworks, and brings about an extremely poor assessment of anomaly detection approaches. To comprehend these issues, they proposed new data set, NSL-KDD, which comprises of selected records of the complete KDD data set [121].

The accompanying are the advantages of the NSL-KDD over the original KDD data set:

1. It excludes excess records in the training set, so the classifiers won't be predisposition towards more regular records.

2. The amount of selected records from every difficulty level gathering is contrarily corresponding to the rate of records in the original KDD data set. Therefore, the classification rates of different machine learning systems fluctuate in a more extensive reach, which makes it more proficient to have a precise assessment of diverse learning techniques.

3. The amount of records in the training and test sets is sensible, which makes it reasonable to run the investigation on the complete set without the need to randomly

select a little portion. Therefore, assessment aftereffects of diverse examination works will be reliable and tantamount.

The NSL-KDD data incorporates 41 features and 5 classes that are normal and 4 sorts of attacks: Dos, Probe, R2L, and U2R.

Denial of Service Attack (Dos) is an attack in which the assailant makes some registering or memory asset excessively occupied or excessively full to handle authentic appeals, or denies legitimate clients access to a machine.

Probing Attack is an endeavour to accumulate data around a system of machines for the obvious reason for evading its security controls.

User to Root Attack (U2R) is a class of endeavour in which the assailant begins with access to a typical client account on the framework (maybe picked up by passwords, a lexicon assault, or social designing) and can misuse some subjection to addition root access to the framework.

Remote to Local Attack (R2L) happens when an assailant who can send packets to a machine over a system yet who does not have a record on that machine abuses some defencelessness to addition nearby get access to as a client of that machine.

5.4 Attributes in KDD CUP99

The data (citing to a bunch of packets over a time length of two seconds, additionally named as packet data, lay down in KDD Cup99 have forty one (41) features. Among these forty one features, 1 to 9 are utilized to illuminate basic features of a packet, 10 to 22 concentrate on content features, 23 to 31 are employed for traffic features with 2 seconds of your moment window and 32 to 41 for host primarily based features.

They are essentially classified into three categories: basic features of individual connection, content features contained by a connection, and traffic features that are computed employing a two seconds time window.

5.4.1 Basic Features

This category includes all the attributes that are extorted from a TCP/IP connection. These features are extorted from the packet header and consist of *src_bytes*, *protocol*, *dst_bytes*, etc.

5.4.2 Content Features

This category is accustomed appraise the payload of the initial communications protocol packet and appears for suspicious behavior within the payload segment. This comprises features like the quantity of failing login makes an attempt, variety of file creation operations etc. Furthermore, the majority of the R to L and U to R attacks do not have any recurrent sequent prototypes. This is often as a result of the very fact that DoS and inquiring attacks involve several connections to some hosts in an exceedingly very short length of your time. However, the R2L and U2R attacks are entrenched inside the information parts of the packets, and customarily involve solely one connection. Therefore to observe these sorts of attacks, content based mostly features are used.

5.4.2.1 Traffic Features

These embrace features that are computed with relevance a window interval and are divided into two classes

i) **"Same host" features:** These characteristics are determined exclusively by researching the connections inside the past a couple of seconds that have consistent end host in light of the fact that the flow association, and figure detail connected with convention conduct, administration and so forth.

ii) **"Same service" features:** These features scrutinize solely the connections within the past a pair of seconds that have constant service just like the current connection. The above of two sorts is referred as "time based traffic features".

Separately from these, there are numerous slow probing attacks that scrutinize the hosts or ports victimization amount larger than two seconds. As a consequence, these forms of attacks do not engender intrusion patterns with a time window of two seconds. To beat this drawback, the "same host" and "same service" options are usually re-computed employing an affiliation window of one hundred connections. These forms of options are referred to as “connection-based traffic features”.

A comprehensive catalog of the set of features distinct for the connection account is given in the following three tables:

Table 5.2 **- Features of KDD CUP99**

Feature Name	***Description***	***Type***
Hot	amount of “hot” signs	Continuous
num failed logins	number of fizzled login endeavours	Continuous
logged in	1 if effectively logged in; 0 overall	Discrete
num compromised	number of "traded off" conditions	Continuous
root shell	1 if root shell is gotten; 0 generally	Discrete
su attempted	1 if "su root" charge endeavored; 0 overall	Discrete
num root	number of "root" gets to	Continuous
num file creations	number of document creation operations	Continuous
num shells	number of shell prompts	Continuous
num access files	number of operations on access control files	Continuous
num outbound cmds	number of outbound commands in an ftp session	Continuous
is hot login	1 if the login belongs to the “hot” list; 0 otherwise	Discrete
is guest login	1 if the login is a “guest” login; 0 otherwise	Discrete

Table 5.3 **- Basic features of individual TCP connections**

Feature Name	***Description***	***Type***
Duration	length (number of seconds) of the connection	continuous
protocoLtype	type of the protocol, e.g. tcp, udp, etc.	Discrete
Service	network service on the destination, e.g., http, telnet, etc.	Discrete
src bytes	number of data bytes from source to destination	continuous
dst bytes	number of data bytes from destination to source	continuous

Flag	normal or error status of the connection	Discrete
Land	1 if connection is from/to the same host/port; 0 otherwise	Discrete
wrong fragment	number of “wrong” fragments	continuous
Urgent	number of urgent packets	continuous

***Table 5.4* - Content features within a connection suggested by domain knowledge**

Feature Name	***Description***	***Type***
Hot	number of “hot” indicators	continuous
num failed logins	number of failed login attempts	continuous
logged in	1 if successfully logged in; 0 otherwise	Discrete
num compromised	number of “compromised” conditions	continuous
root shell	1 if root shell is obtained; 0 otherwise	Discrete
su attempted	1 if “su root” command attempted; 0 otherwise	Discrete
num root	number of “root” accesses	continuous
num file creations	number of file creation operations	continuous
num shells	number of shell prompts	continuous
num access files	number of operations on access control files	continuous
num outbound cmds	number of outbound commands in an ftp session	continuous
is hot login	1 if the login belongs to the “hot” list; 0 otherwise	Discrete
is guest login	1 if the login is a “guest” login; 0 otherwise	Discrete

***Table 5.5* - Traffic features computed using a two-second time window**

Feature Name	***Description***	***Type***
Count	number of connections to the same host as the current connection in the past two seconds	continuous
Note: The following features refer to these same-host connections		

serror_rate	% of connections that have “SYN” errors	Continuous
rerror_rate	% of connections that have “REJ” errors	Continuous
same_srv_rate	% of connections to the same service	Continuous
diff_srv_rate	% of connections to different services	Continuous
srv_count	number of connections to the same service as the current connection in the past two seconds	Continuous
Note: The following features refer to these same-service connections		
srv_serror_rate	% of connections that have “SYN” errors	Continuous
srv_rerror_rate	% of connections that have “REJ” errors	Continuous
srv_diff_host_rat	% of connections to different hosts	Continuous

5.5 Symbolic Features

The KDD’99 dataset has seven symbolic options out of that four are binary features and other three have more than two attributes. The later three features for symbolic conversion will be converted into numerical features in our work. These symbolic features are listed in table below along with their number of attributes.

Table 5.6 **- Symbolic Features**

No	Feature Name	Number of attributes
1	Protocol Type	3
2	Service	70
3	Flag	11
4	Land	2
5	Logged in	2

6	is_hot_login	2
7	is_guest_login	2

Protocol

The protocol feature describes the protocol type for the connection. There are three types of protocols which are represented symbolically in the KDD’99 dataset as follows

- TCP
- UDP
- ICMP

Service

The service feature describes different types of services that are available for connections to utilize. There are 70 services that can be used. Each of the 70 service has been grouped into eight clusters depending on its utilization by TCP port [1].

- Services that are used to remotely access other machines.
- File transfer services e.g., *ftp*.
- Mail transfer services e.g. *smtp*.
- Web Services like web server, *http*.
- Services used to obtain statistics of the system.
- Name servers services.
- Services for other protocols like ICMP.

Flag

The flag features describe the status of the connection. There are 13 flags but 11 of them are used in KDD’99 dataset which are S0, S1, SF, REJ, S2, S3, RSTO, RSTR, RSTOS0, SH and OTH. These features are further clustered into six types which are described in table below.

Table 5.7 - **Flag Features [1]**

Cluster	**Name**	**Description**
F1	S0	Connection attempt seen, no reply
	REJ	Connection attempt rejected
F2	S1	Connection established but not terminated
	SF	Regular establishment and termination
	OTH	No SYN seen, just midstream traffic
F3	S2	Connection established and close attempt seen by originator
	RSTO	Connection established, originator aborted
F4	S3	Connection established and close attempt seen by responder
	RSTR	Connection established, responder aborted
F5	RSTOS0	Originator sent a SYN followed by a RST, SYN ACK not seen by the responder
	SH	Originator sent a SYN followed by a FIN, SYN ACK not seen by the responder
F6	RSTRH	Responder sent a SYN ACK followed by a RST, SYN not seen by the originator
	SHR	Responder sent a SYN ACK followed by a FIN, SYN not seen by the originator

These were the symbolic features (Flag, Protocol and Service) included in KDD’99 dataset. The other 4 numeric features which are Land, Logged-in, is_hot_login and is_guest_login have binary values 0 and 1. These binary features are given above in Table 5.7. The three symbolic features protocol, service and flag are converted to numeric features using symbolic conversion methods. Below is the specification of four numeric binary features of KDD’99 dataset.

Land

The land feature specifies whether a connection is from or to the same host (port) or to the different host (port). Its value is 1 if the connection is from or to the same host (port) and it is 0 otherwise [9].

logged_in

The logged_in feature specifies the success of a log in attempt by a user to the network or system. Its value is 1 if a user has been successfully logged in and its value is 0 if user fails to log in [9].

is_hot_login

The is_hot_login feature tells whether a login is from the “hot” list of disc. Its value is 1 if login is from the list else 0[5].

is_guest_login

The is_guest_login feature tells whether a login is from the “guest” account. Its value is 1 if login is a “guest” and 0 otherwise [5].

5.6 Numeric features

The KDD’99 dataset has 34 numeric features with different ranges. We shall discuss each of the numeric features.

Duration

The duration feature gives information about the time period of the connection in seconds. The range of the duration varies from 0 to 58329 [12].

src_bytes

This feature tells about the amount of the data in bytes sent from the source of the network connection to the destination. Its value ranges from 0 to 1.3 billion [5]. This large range was converted to a small range within 0.00 to 10.00 in our processing be applying logarithmic scaling technique.

dst_bytes

This feature tells about the amount of the data in bytes sent from the destination of the network connection to the source of the network connection. Its value also ranges from 0 to 1.3 billion [5]. This large range was also converted to a small range within 0.00 to 10.00 in our processing be applying logarithmic scaling technique.

wrong_fragment

This feature specifies the number of wrong fragments in a network connection. Its value ranges from 0 to 3.

urgent

This feature specifies the number of urgent packets in a network connection. Its value ranges from 0 to 14.

hot

This feature specifies the number of "hot" indicators in a network connection [9]. Its value ranges from 0 to 101.

num_failed_logins

This feature specifies the number of login attempts that have been failed. These failed login attempts ranges from 0 to 5.

num_compromised

This feature specifies the number of "compromised" conditions in a network connection [9]. These number of "compromised" conditions ranges from 0 to 9.

root_shell

This feature specifies whether the root_shell logged in. If root_shell has logged in , its value is 1, otherwise 0 [9].

su_attempted

This feature specifies whether the su_root command has been executed. If su_root command has been executed, its value is 1 and 0 otherwise.

num_root

This feature specifies the number of root accesses to the network. Its value ranges from 0 to 7468 [11].

num_file_creations

This feature specifies the number of operations for file creations. These operation ranges from 0 to 100 [11].

num_shells

This feature specifies the number of prompts in shell [9]. Its value ranges from 0 to 5.

num_access_files

This feature specifies the number of operations performed on access control files [9]. Its value ranges from 0 to 9.

num_outbound_cmds

This feature specifies the number of outbound commands executed in an ftp session [9].

count

This feature specifies the number of connections with the same host as the current connections had, that have been occurred in past two seconds [9]. Its value ranges from 0 to 511.

srv_count

This feature specifies the number of connections with the same service as the current connections had, that have been occurred in past two seconds [9]. Its value ranges from 0 to 511.

serror_rate

This feature specifies the percentage of the network connections that contains "SYN" errors [12].

srv_serror_rate

This feature specifies the percentage of the network connections that contains "SYN" errors [12].

rerror_rate

This feature specifies the percentage of the network connections that contains “REJ” errors [12].

srv_rerror_rate

This feature specifies the percentage of the network connections that contains “REJ” errors [12].

same_srv_rate

This feature specifies the percentage of the network connections that are dedicated to same service [12].

diff_srv_rate

This feature specifies the percentage of the network connections that are dedicated to different services [12].

srv_diff_host_rate

This feature specifies the percentage of the network connections that are dedicated to different hosts [12].

dst_host_count

This feature specifies the number of connections to the destination host as the current connections had, that have been occurred in past two seconds [12]. Its value ranges from 0 to 255.

dst_host_srv_count

This feature specifies the number of connections to the destination service as the current connections had, that have been occurred in past two seconds [12]. Its value ranges from 0 to 255.

dst_host_same_srv_rate

This feature specifies the percentage of connections that have been initiated to the same service at the destination host [12].

dst_host_diff_srv_rate

This feature specifies the percentage of connections that have been initiated to the different services at the destination host [12].

dst_host_same_src_port_rate

This feature specifies the percentage of connections that have been initiated to the same source ports at the destination host [12].

dst_host_srv_diff_host_rate

This feature specifies the percentage of connections that have been initiated to the different hosts at the destination host [12].

dst_host_serror_rate

This feature specifies the percentage of the network connections that contains "SYN" errors at the destination host [12].

dst_host_srv_serror_rate

This feature specifies the percentage of the network connections that contains "SYN" errors at the destination host [12].

dst_host_rerror_rate

This feature specifies the percentage of the network connections that contains "REJ" errors at the destination host [12].

dst_host_srv_rerror_rate

This feature specifies the percentage of the network connections that contains "REJ" errors at the destination host [12].

5.7 Classification of Attacks

The knowledge set in KDD Cup99 have traditional and twenty two attacks kind data with forty one attributes and Table 5.8 shows few knowledge set. All generated traffic patterns finish with a label either as 'normal' or any form of 'attack' for future analysis.

Table 5.8 **- Classification of Attacks**

Feature Name	***Packet-1*** ***(normal)***	***Packet-2*** ***(Neptune)***
Duration	0	0
protocol_type	Tcp	Tcp

Service	http	Private
Flag	SF	REJ
src_bytes	327	0
dst_bytes	467	0
Land	0	0
wrong_fragment	0	0
Urgent	0	0
Hot	0	0
num_failed_logins	0	0
logged_in	1	0
num_compromised	0	0
root_shell	0	0
su_attempted	0	0
num_root	0	0
num_file_creations	0	0
num_shells	0	0
num_access_files	0	0
num_outbound_cmds	0	0
is_hot_login	0	0
is_guest_login	0	0
Count	33	136
srv_count	47	1

serror_rate	0.00	0.00
srv_serror_rate	0.00	0.00
rerror_rate	0.00	1.00
srv_rerror_rate	0.00	1.00
same_srv_rate	1.00	0.01

There are numerous types of attacks that are inflowing into a network over a length of time and therefore the attacks are classified into the subsequent four main categories.

- Denial of Service (DoS)
- User to Root (U2R)
- Remote to User (R2L)
- Probing

5.7.1 Denial of Service

Denial of Service is a category of attacks where associate degree attacker makes some computing or memory resource too busy or too full to handle justifiable requests, denying legal users access to a device. The various ways to launch a DoS attack area unit are:

- By abusing the computer is legitimate options
- By targeting the implementation bugs
- By utilizing the misconfiguration of the machines

DoS attacks area unit classified supported the services that associate degree assailant renders untouchable to legitimate users.

5.7.2 User to Root

In User to Root attack, associate degree assailant starts with access to a traditional user account on the system and gains root access. Regular programming mistakes associate degraded atmosphere assumptions provide an assailant the chance to take advantage of the vulnerability of root access.

5.7.3 Remote to User

In Remote to User attack, associate degree assailant sends packets to a machine over a network that exploits the machine's vulnerability to realize native access as a user illicitly. There are contradictory types of R2L attacks and therefore the most typical attack during this category is completed by victimization social engineering.

5.7.4 Probing

Probing may be a category of attacks wherever associate degree assailant scans a network to collect information so as to seek out identified vulnerabilities. Associate degree assailant with a map of machines and services that are out there on a network will manipulate the data to appear for exploits. There are differing types of probes: a number of them misuse the computer's legitimate options and a few of them use social engineering techniques. This category of attacks is that the most typical as a result of it needs little or no technical experience.

Table 5.9 **- Effects of Different Kind of Attacks**

Name of the Attack	**Type**	**Mechanism**	**Effect of the Attack**
Back	DoS	Misuse/Bug	Slows down server response
Land	DoS	Bug	Slows down server response
Neptune	DoS	Misuse	Slows down server response

Smurf	DoS	Misuse	Slows down the network
Pod	DoS	Misuse	Slows down server response
Teardrop	DoS	Bug	Reboots the machine
Loadmodule	U TO R	Poor environment sanitation	Gains root shell
buffer_overflow	U TO R	Misuse	Gains root shell
Rootkit	U TO R	Misuse	Gains root shell
Perl	U TO R	Poor environment sanitation	Gains root shell
Phf	R2L	Bug	Executes commands as root
guess_passwd	R2L	Login misconfiguration	Gains user access
Warezmaster	R2L	Misuse	Gains user access
Imap	R2L	Bug	Gains root access
Multihop	R2L	Misuse	Gains root access
ftp_write	R2L	Misconfiguration	Gains user access
^{s}py	R2L	Misuse	Gains user access
Warezclient	R2L	Misuse	Gains user access

Satan	Probe	Misuse of feature	Looks for known vulnerabilities
Nmap	Probe	Misuse of feature	Identifies active ports on a machine
Portsweep	Probe	Misuse of feature	Identifies active ports on a machine
Ipsweep	Probe	Misuse of feature	Identifies active machines

The different sorts of attack, their mechanism and consequences [116] area unit listed in above table. The unauthorized persons principally misuse the network or system in numerous manners and gain the network access or bog down the response.

Although several irregularities are existed in KDD Cup 99 data set, analysis activities in IDS area unit still are using the KDD Cup 99 dataset for analysing and exploring new approaches for higher IDS. Hence, the projected technique has been experimented and analysed with KDD Cup 99.

5.8 Summary

This chapter outlines the structure of the dataset utilized in the research work. The various kinds of features like distinct and continuous features are considered with attention on their role within the attack. The attacks are classified with a short introduction to each one. The principle explanation behind selecting KDD Cup 99 dataset is that right now, it is the generally utilized exhaustive information set that is imparted by numerous analysts. In this dataset, 41 characteristics (Table 5.2) are utilized as a part of each one record to describe network traffic behaviour. Around these 41 traits, 38 are numeric and 3 are typical. Characteristics show in KDD information set is aggregated into three classifications. The *protocol_type*, *service*, *flag*, *land*, *logged_in, is_hot_login*, and *is_guest_login* are marked as disconnected or categorical features and other 34 characteristics are named as continuous features.

One of the real confinements of this methodology is that it is costlier and it does not think about new ambushes. Along these lines, this proposed exploration work utilizes the benchmark dataset extracted for the 1999 KDD intrusion detection challenge, by MIT Lincoln Labs [7, 8]. The primary point of interest of utilizing this dataset is that the proposed examination work is equipped for furnishing huge information that are effortlessly imparted to different specialists and engineers.

CHAPTER 6
IMPLEMENTATION, EXPERIMENTS AND RESULTS

This Chapter implements the network intrusion detections system discussed throughout the thesis and explains in detail the steps used in the implementation. Furthermore, the results are evaluated and compared to arrive at appropriate findings as discussed in the following sections.

This section portrays the procedure of Network Intrusion detection systems utilizing two machines learning techniques on the dataset that is pre-processed utilizing two distinctive symbolic conversion techniques. The following figure shows the System Architecture of the Research:

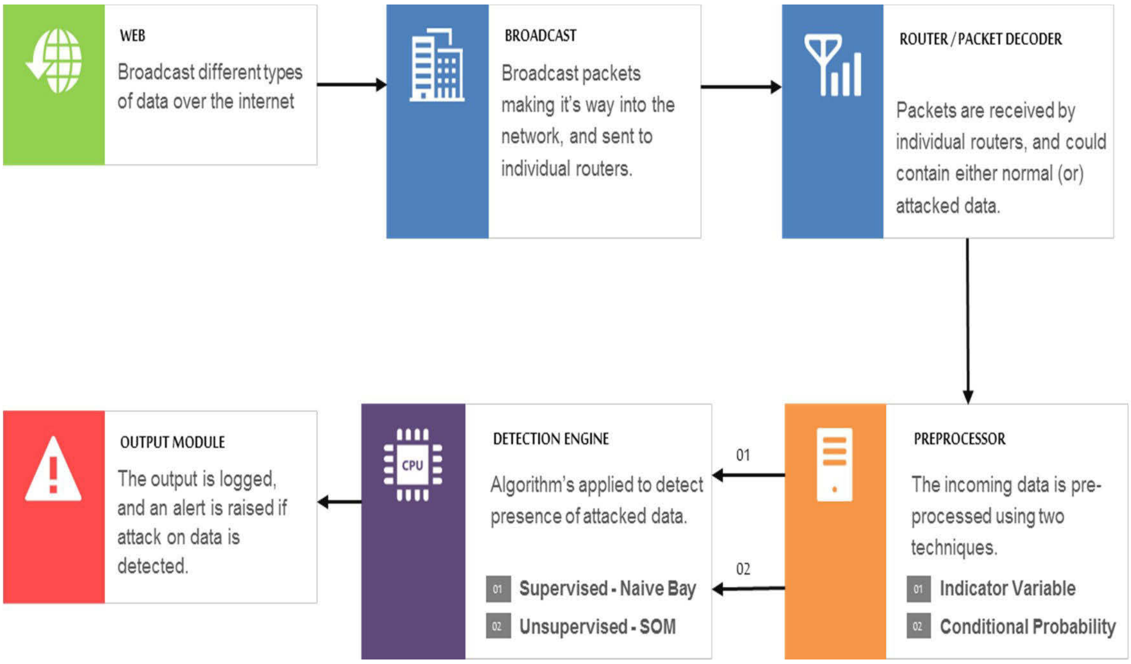

WEB
Broadcast different types of data over the internet
BROADCAST
Broadcast packets making it's way into the network, and sent to individual routers.
ROUTER / PACKET DECODER
Packets are received by individual routers, and could contain either normal (or) attacked data.
PREPROCESSOR
The incoming data is pre-processed using two techniques.
01 Indicator Variable
02 Conditional Probability
01
02
DETECTION ENGINE
Algorithm's applied to detect presence of attacked data.
01 Supervised - Naive Bay
02 Unsupervised - SOM
CPU
OUTPUT MODULE
The output is logged, and an alert is raised if attack on data is detected.

Figure 6.1 **- System Architecture**

The above diagram is an overview of how network intrusion can be detected using the proposed techniques. Different types of data that are broadcasted over the internet will make its way through different routers before entering the network. This data is in the form of packet whose attributes have been discussed in Chapter 5. These packets are received by IDS which will further pre-process the data that may or may not be attacked. The pre-processing takes place using two techniques – Indicator Variable and Conditional Probability. After converting the data into its appropriate forms, two types of machine learning algorithms will be applied for detection. These techniques are classified as supervised or unsupervised as discussed in Chapter 4. As mentioned before, this thesis will use Naïve Bayes algorithm for supervised learning technique and Self-Organising Maps for unsupervised learning technique. These algorithms will detect network intrusion on the data and segregate the attached data from the un-attacked data. The results are further logged and an alert is raised if the attack on data is identified.

A short portrayal of the steps that are used in this research work is given below,

- Collection of the training and test dataset from NSL-KDD dataset [7].
- Conversion of ASCII qualities to numeric qualities to be stacked in MATLAB
- Implementation of two diverse dataset pre-processing calculations.
- Implementation of Machine Learning Algorithms.
- Training the Machine Learning Algorithms on pre-processed dataset.
- Testing the Machine Learning Algorithms on the test dataset.
- Comparison of results.

6.1 Collection of the Dataset

Before NSL KDD information set the majority of the examiners or analysts utilized KDD'99 information set for the examination or detection of the intrusion, yet the conclusion of the KDD'99 information could not fulfil to the specialist or scientists. There are numerous issues in KDD'99 information set which has been overcome by NSL KDD information set. NSL-KDD is a dataset proposed by Tavallaee et al. [7]. NSL-KDD dataset is a lessened form of the first KDD’99 dataset. NSL-KDD comprises of the same characteristics as KDD’99. The KDD99 dataset comprises of 41 characteristics and one class trait. The class characteristic has 21 classes that fall under four sorts of strike: Probe assaults, User to Root (U to R) attacks, Remote to Local (R2l) attacks and Denial of Service (Dos) attacks. This dataset has a twofold class characteristic. Additionally, it has a sensible number of preparing and test cases which make it common sense to run the probes.

The NSL-KDD information set lays the accompanying points of interest over the first KDD information set.

1. NSL KDD information set does not incorporate repetitive records in the train set, so the classifiers will not be inclined towards more frequent records.

2. There are no repeated records in the proposed test sets; consequently, the execution of the learners is not inclined by the techniques which have better detection rates on the continuous records.

3. The number of chosen records from every challenge level assembly is conversely relative to the rate of records in the first KDD information set. Subsequently, the classification rates of unique machine learning routines change in a more extensive extent, which makes it more proficient to have a precise assessment of different learning procedures.

4. The amount of records in the training and test sets is reasonable, which makes it moderate to run to probe the complete set without the requirement of haphazardly select a little partition. Thus, assessment effects of different examinations will be steady and equivalent.

The training dataset is chosen from NSL-KDD dataset to incorporate 500 connection vectors out of which 250 are attack and 250 are normal connections. A small number of vectors are picked for preparing dataset as the transforming in MATLAB R2013A is slow. The test dataset likewise incorporates 500 connection vectors. There are numerous ASCII values in the dataset speaking to ordinary connection with the statement "normal" and assault connections with the saying "attack". The statement "normal" is displaced by the number 0 and the saying "attack" is reinstated by the number 1.

6.2 Data Processing

As intrusion detection system is an acute component of secure information systems scrutinize all data features to detect intrusion. Some of the features may be redundant or be different scales therefore they need pre-processing. NSL KDD data set have a different connection record all connection record have total 41 features. 38 of these features are numeric and three of them are symbolic thus, we map symbolic-valued attributes to numeric-valued attributes. Symbolic features like Protocol type (three different symbols – *tcp, udp, icmp*), Service (70 different attributes), and Flag (11 different symbols). They were mapped to integer values ranging from 1 to *N* where *N* is the number of symbols.

NSL KDD information set have two sorts of information set changed over to numeric qualities utilizing two separate methodologies which are Conditional Probability and Indicator Variable Symbolic Conversion system. It has distinctive sorts of field attributes. These traits have some numeric fields and some symbolic type of fields. These fields are firstly changed over into a numeric structure. According to this, first line is chosen and discovers max value attribute, and these max worth is isolated into four equivalent interims. These interims characterize class esteem. There are four classes, to be specific: very low, low, medium and high. In this rationale, the information is changed over into a downright structure and this is demonstrated as,

Very squat	Low down	Medium	High
Class A	Class B	Class C	Class D

1 M/4 2*M/4 3*M/4 M

6.2.1 Indicator Variables

The fundamental methodology of indicator variables is that, *1* shows the event of classifications of characteristics and *0* demonstrates its non-occurrence of classes of characteristics. The ostensible characteristic *X* speaks to the characteristics with *N* notable classes, a set of *N* indicator variables might be produced based upon the categories exist in the qualities of information set [117]. In the exploration nine essential characteristics are utilized yet after applying this strategy, it grows and it is changed into 122 properties. This method is applied on substantial number of categories of traits, for example, *Protocol*, *Service* and *Flag* characteristics. For example, if an attribute has three qualities, that attribute might be articulated to by *001,010,100 or 100,010,001* [118].

The point when the KDD Cup 1999 dataset is pre-processed utilizing indicator variables transformation, the 41 characteristics of every connection vectors are expanded to 122 characteristics as three typical traits which are *Protocol* (3 properties), *Service* (70 features) and *Flag* (11 features) are traded by their indicator variables vector each.

Illustration

Protocol feature have three separate traits:

$$Protocol\ (F2) = \{tcp,\ udp,\ icmp\}.$$

The three traits are traded by indicator variables as takes after:

Table 6.1 **- Sample Dataset for Indicator Variable Conversion**

f1	**f2**	**…………………**	**f41**	**Attack/ Normal**
X1	Udp	…………………	Y1	Normal
X2	Tcp	…………………	Y2	Normal
X3	Udp	…………………	Y3	Attack

$\text{udp} \ = [0\ 0\ 1]$

$\text{tcp} \ = [0\ 1\ 0]$

$\text{icmp} = [1\ 0\ 0]$

Table 6.2 **- Converted Dataset after Indicator Variable Conversion**

f1	**f21**	**f22**	**f23**	**…………………**	**f41**	**Attack/ Normal**
X1	0	0	1	…………………	Y1	Normal

X2	0	1	0	…………………	Y2	Normal
X3	0	0	1	…………………	Y3	Attack

The dataset is pre-processed in the testing stage utilizing the same strategy which has been utilized in the training phase.

6.2.2 Conditional Probability

The second strategy is to change over a typical characteristic with an exhibit of conditional probability for getting each one class given the trait that has specific typical quality. Thus, every typical vector x_k of a characteristic "*a*" may be swapped by the accompanying N-dimensional vector of conditional probabilities [118], [117]. *n* is the amount of classes of the training set and *m* is the number of classifications of the typical vector of x_k. We have effectively connected the system on typical characteristics of information set.

This exhibit is given as:

$$[P(C_1 \mid f = a_i), P(C_2 \mid f = a_i), ..., P(C_n \mid f = a_i)] \; \forall i = 1......m \,, \tag{6.1}$$

where *C* is the class given that a particular feature *f* has characteristics *a*, *m* is the amount of properties of a characteristic and *n* is the amount of classes which are two for our situation i.e., normal and attack.

The Conditional Probability Conversion methodology is applied to typical characteristics of KDD'99 dataset. These characteristics incorporate Protocol, Service and Flag characteristic. The point when the KDD Cup 1999 dataset is pre-processed utilizing conditional probability change, the 41 characteristics of every connection vectors are expanded to 44 characteristics as three typical traits which are *Protocol* (three properties), *Service* (70 properties) and *Flag* (11 properties) are replaced by their conditional probability vector each. Each of the conditional probability vectors holds two qualities. Let's have a sample of Protocol Feature to get in depth and more clear understanding of the Conditional Probability approach.

Example

Protocol feature have three separate qualities:

$$Protocol\ (F2) = \{tcp,\ udp,\ icmp\}$$

The three qualities are supplanted by conditional probabilities as accompanies:

Table 6.3 **- Sample Dataset for Conditional Probability Conversion**

f1	**f2**	**…………………**	**f41**	**Attack/ Normal**
X1	Udp	…………………	Y1	Normal
X2	Tcp	…………………	Y2	Normal
X3	Udp	…………………	Y3	Attack

By applying the conditional probability conversion technique, we get the accompanying effects:

$udp\ = [P(Normal|f2 = udp),\ \ P(Attack|f2 = udp)]$

$tcp\ \ = [P(Normal|f2 = tcp),\ \ P(Attack|f2 = tcp)]$

$icmp = [P(Normal|f2 = icmp), P(Attack|f2 = icmp)]$

Table 6.4 **- Converted Dataset after Conditional Probability Conversion**

f1	**f21 (Normal)**	**f22 (Attack)**	**…………………**	**f41**	**Attack/ Normal**
X1	½	½	…………………	Y1	Normal

X2	1	0	…………………	Y2	Normal
X3	½	½	…………………	Y3	Attack

6.3 Training and Testing Phase

Both supervised and unsupervised machine learning techniques are prepared on the pre-processed training dataset. The Naïve Bayes calculation is utilized as supervised machine learning method and SOM is utilized as unsupervised machine learning strategy. In the wake of preparing both the systems for two separate sorts of pre-processed dataset, their executions are looked at in recognizing the surprise attack connections.

6.3.1 Self-Organizing Map

SOM is essentially used to discover the likeness guide of the information. There are typically three sorts of topologies utilized as a part of the SOM mapping, which is rectangular, hexagonal and irregular mapping framework. The separation is found from typically four separate routes like Euclidean, box, link and manhattans distance. The fundamental SOM calculation is condensed beneath

1. Firstly set the topology of neurons for SOM mapping.

2. The information $''x''$ is likewise select haphazardly from given information set however it takes diverse extent of characteristics set so firstly transpose the first information.

3. The SOM essential intention is to discover winning neuron; it is likewise called winner neuron $''h''$ which is focus through distinctive separation capacity like it could be indicated in mathematical statement 6.2 which is assembled by Euclidean separation.

$$||W_h - x|| = min||W_1 - x||. \qquad (6.2)$$

4. All topology neurons $''ij''$ in the area of winning neuron $''h''$ which is determine through all weight vectors by normally utilized Gaussian capacity apply between the

two neurons separate in yield layer. This capacity (j,h) speaks to the connection of neuron j and h which is nearly identified with one another. It is focused by equation6.3;

$$\|W_h - x\| = min\|W_1 - x\|. \tag{6.3}$$

Learning utilizing SOM obliges instatement of SOM as codebook of neurons, preparing of SOM as stated by the calculation examined above and at long last the test stage. The depiction of each of the phase is given below.

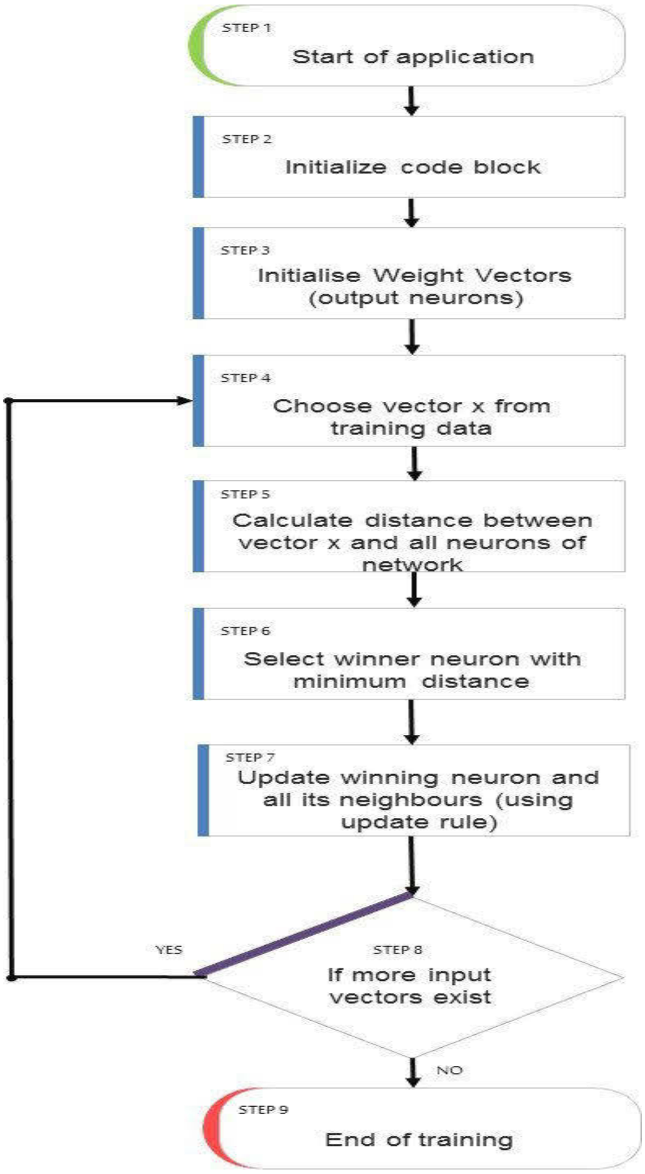

STEP 1
Start of application
STEP 2
Initialize code block
STEP 3
Initialise Weight Vectors (output neurons)
STEP 4
Choose vector x from training data
STEP 5
Calculate distance between vector x and all neurons of network
STEP 6
Select winner neuron with minimum distance
STEP 7
Update winning neuron and all its neighbours (using update rule)
STEP 8
If more input vectors exist
YES
NO
STEP 9
End of training

Figure 6.2 **- Learning Phase of SOM**

6.3.1.1 Codebook Initialization

The codebook array in the code speaks to the neurons of the self-organising map. The codebook is chosen to have 30 by 30 dimensions. There will be 900 neurons in the codebook vector, each with dimensions same as the input instance. The input instance has dimensions according to the conversion techniques applied on the dataset. The point when conditional probability conversion strategy is applied, a data set has 44 dimensions; subsequently codebook neurons additionally have 44 dimensions. Correspondingly, when indicator variable conversion method is applies, an input vector has 122 dimensions; henceforth, codebook neurons likewise have 122 dimensions. The codebook neurons are initialised utilizing the rand function of MATLAB R2013A

6.3.1.2 Training Phase

The self-organizing map is trained on 500 connection vectors taken from NSL-KDD dataset. These connection vectors are classified as normal and attack. The SOM is trained on the dataset pre-processed by two different conversion methods which are Indicator Variable and Conditional Probability and have been discussed in section 6.2. The SOM was selected to have 35 by 35 dimensions. The dataset was trained in 15 iterations. The neighbourhood radius of SOM was chosen to be 15. The learning rate *α(t)* is taken to be 0.1 at the start and it decreases with time in each epoch. Mexican hat was chosen as the update rule.

$$h(\rho,t) = \exp\left(-\frac{\rho^2}{\sigma^2(t)}\right)\left(1-\frac{2}{\sigma^2(t)}\rho^2\right), \tag{6.4}$$

where *h(ρ,t)* is the Mexican hat function, *ρ* is the distance of the neighbours to the winner neuron which is calculated using Euclidean distance formula and *σ* is the space of the neighbourhood which is 15 at the start which is decreased with each epoch.

After the calculation of update rule for each neighbourhood, the neighbours are updated with the formula given as

$$W_{ij}(t+1) = W_{ij}(t) + \alpha(t)h(\rho,t)(X(t) - W_{ij}(t)), \tag{6.5}$$

where $W_{ij}(t)$ is the weight vector of SOM neuron at time t and $X(t)$ is the input vector at time *t*.

The pseudo code for training phase of self-organising map is as follow:

```
learn(Samples, MapNodes, iterations) {
    for(iteration =0 to iterations) {
        sample = getRandomSample(Samples)
        bmu = undefined, bmuDistance = inf
        foreach(mapNode in MapNodes) {
            distance = euclideanDistance(sample, mapNode)
            if(distance <bmuDistance) {
                bmu = mapNodebmuDistance = distance}
        }
        foreach(mapNode in MapNodes ) {
            bmuDistance = euclideanDistance(bmu, mapNode)
            alterMapNode(mapNode, sample,
            getNeighborhoodFactor(iteration, bmuDistance),
            getLearningRateFactor(iteration))
        }
    }
}
```

The training phase completes in 15 iterations for 500 inputs each. During each iteration, the codebook vector is initialized according to the weights of the neurons. After 15 iterations we get an updated self-organizing map which is used in the testing phase.

6.3.1.3 Testing Phase

In the stir of getting yield of training SOM, re-enact the testing or unseen yield information is put on prepared system. After get record of winning neurons of preparing and testing information set. It is identified with both distinctive systems for the characterization of normal and attack connections of dataset. The new inputs from the test dataset are provided for the framework to figure out if it is an attack connection or normal connection.

In the testing phase, the dataset is taken to have 500 connection vectors among which half are normal and half are attack. The new inputs from the test dataset are given to the system to determine whether it is an attack connection or normal connection.

In testing using SOM, the updated codebook vector is used for testing each connection vector. The Euclidean Distance is taken from each connection vectors to all neurons in the SOM. The neuron with minimum distance to the input vector is marked as winner neuron and then verified if it has classified the input correctly.

To calculate the distance of the input vector to SOM to find the best matching unit, commonly used distance formula is Euclidean Distance. It is given as:

$$d_{ij} = \sqrt{(x_1 - w_{ij1})^2 + (x_2 - w_{ij2})^2 + \cdots + \left(x_n - w_{ijn}\right)^2}, \tag{6.6}$$

where d_{ij} is the distance of input neuron x of n dimensions to the w_{ij} of the output layer in the SOM;

i and j are the coordinates of the weight vector on the map.

The SOM neuron with the minimum distance to the input neuron is designated as winner neuron $d(k_1,k_2)$, where k_1 and k_2 are indices of winner neuron.

$$d(k_1, k_2) = \min_{i,j} d_{ij}. \tag{6.7}$$

Then the number of false positives (attacks classified as normal) and false negative (normal classified as attacks) are determined. Finally the accuracy of the classification using SOM is determined.

Following pseudo code shows how the grouping is carried out in our implementation.

SOM testing stage (pseudo code):

```
SOM_Classify(Samples, MapNodes, UpdatedCodebookSamples) {
    correct_count = 0, success_rate = 0
    foreach(sample in Samples) {
        bmuDistance = inf, bmu = undefined
        foreach(mapNode in MapNodes) {
            distance = euclideanDistance(sample, mapNode)
            if(distance <bmuDistance) {
                bmu = mapNodebmuDistance = distance
            }
        }
    sample.x = bmu.x  ;sample.y = bmu.y
}
```

In testing utilizing SOM, the upgraded codebook vector is utilized for testing every association vector. The Euclidean separation is taken from every association vectors to all neurons in the SOM. The neuron with least separation to the data vector is checked and figured out whether it has characterized the information accurately. At that point, the amount of false positives (strike considered ordinary) and false negative (typical considered attacks) are dead set. At long last the correctness of the characterization utilizing Self Organizing Map is determined.

6.3.2 Naïve Bayes

Naïve Bayesian accepts that the impact of property estimation on a given class is autonomous of the qualities of alternate traits. This supposition is called class conditional autonomy. It is made to rearrange the calculations included and, in this sense, think about "Guileless". Gullible Bayesian permits the representation of conditions around subsets of properties [9]. Though the utilization of Bayesian systems lies turned out to be successful in specific circumstances, the outcomes acquired are exceptionally subject to the suspicions about the conduct of the target framework, thus a deviation in these theories prompts discovery slips, attributable to the model recognized [9].

Naïve Bayes Classification is a supervised learning technique. In supervised learning the aim is to train a system to map the input to output given the correct values are provided by the supervisor [9]. Naïve Bayes Classifier is based on Bayesian Classification technique. Bayes rule calculate the posterior probability *P(C|x)* using likelihood *P(x|C)* and prior *P(C)* with evidence *P(x)* as below:

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}, \tag{6.8}$$

where *C* is the class and *x* is the input.

Naïve Bayes Classifier is a Bayesian Network which is based on the assumption that all the input attributes are conditionally independent given the target value [9]. Given a series of *n* attributes, the Naïve Bayes classifier make *2n!* independent assumptions [6]. It reduces a multivariate problem to a group of univariate problem. In Naïve Bayes a new instance is provided with a tuple of *n* attribute values ($a_1, a_2, \ldots\ldots, a_n$), where *n* is the dimension of input instance.

$$\prod_i P(a_i|v_j) = P(a_1, a_2, \ldots.., a_n|v_j), \tag{6.9}$$

$$v_{nb} = \arg\max_{vj\epsilon v} P(v_j) \prod_i P(a_i|v_j), \tag{6.10}$$

where v_{nb} denotes the output value generated by the Naïve Bayes Classifier;

v_j is the target value that can be taken by the new instance from the set *V*[1];

P(v_j) is the probability of target value;

P(a$_i$|v$_j$) is the conditional probability that a particular feature *f* has attribute *a* given the target value *v*.

In our case:

$$v = \begin{cases} 0 \ connection & = normal \\ 1 \ connection & = attack \end{cases} . \tag{6.11}$$

There is no industriousness of information here, implying that each run of the project re-takes in everything and afterward approves once more. The training set is randomized and part into a subset of preparing specimens and a subset utilized a new instance of an input. By re-finishing the entire methodology starting with no outside help like this, we can do numerous races to decide out the likelihood that one run was simply a sporadic "best case". Obviously, in a generation framework, the learning gained from a training set is persevered and reused later.

Learning using Naïve Bayes algorithm requires training phase and the test phase. The description of each of the phase is given below.

6.3.2.1 Training Stage

Naïve Bayes Classifier is likewise prepared on 500 connection vectors which are characterized into two classes Normal and Attack. In preparing period of Naïve Bayes, *P(v$_j$)* and *P(a$_i$|v$_j$)* are figured for every connection vector in the training dataset. The learning calculation is prepared on two kind of pre-processed dataset. In both cases, a vector is made for each one characteristic. The measure of this vector is twofold the properties of a characteristic. The explanation for the twofold size of the vector is that every component of the vector will hold the conditional probability quality of each one characteristic of the characteristic given the class is typical or strike. In the first place half partition of the vector is involved with the conditional probability of the properties of the characteristic given the class is normal and the other half holds the conditional probability of characteristics given the class is attack.

The pseudo code for this stage is indicated below. A subset of the example information is nourished to this normal, as depicted previously.

Pseudo code for the Naïve Bayes preparing stage:

```
learn_samples(Samples, Categories) {
    foreach( sample in Samples ) {
      sample.category.samples += sample
      foreach( term in sample.terms ) {
          sample.category.termcount[term] += 1
          total.termcount[term] += 1
          }
     }
     foreach(category in Categories) {
          category.priori= count(category.samples)/ count(samples)
          foreach(term in category.termcount) {
               category.conditional_prob[term]=category.termcount[term]/total.termcount[term]
          }
     }
}
```

In its exposed being, what this piece of the calculation does is to compute the posterior probability for every classification, and the conditional probability for each one characteristic in every class.

Note that, in above pseudo code we elude "a specimen's class" (sample category). Despite the fact that this may come out to be irrational (the specimen knows its category!), recollect that these are pre-categorized examples used to prepare the classifier.

6.3.2.2 Testing Phase

In testing stage, the same dataset is picked for Naïve Bayes as is picked for SOM having 500 connection vectors. The vectors of conditional probability of qualities that were saved in preparing period of Naïve Bayes are utilized as a part of testing stage. For each one input vector from the test dataset, first the probability is found that the connection is a normal, and afterward the probability is found that the connection is attack. Both the probabilities discovered and contrasted to figure out which one is greatest. In the event that the probability for normal connection is more excellent than attack connection, the information is considered normal else attack. So as to focus the probability, the item is taken of the probabilities found in preparing stage for each one characteristic of a characteristic.

The testing phase of the project is the part that really lets us know whether our classifier is acting as it should. It utilizes the remnant of the example information after the training part has done its work, and is indicated as pseudo code.

Pseudo code for the Naïve Bayes check stage

```
classify_samples(Samples, Categories) {
    max_posterior = 0 best_category = undefined correct_count = 0, success_rate = 0
    foreach(sample in Samples) {
        foreach(category in Categories) {
            posterior = calculate_posterior(sample, category)
            if(posterior >max_posterior)
                { best_category = category
                max_posterior = posterior
9                               }
10              }
11              if(best_category == sample.category) correct_count++
12      }
13      success_rate = correct_count / count(samples)
14}
```

You may have recognized that maybe the most intriguing a piece of this component, specifically the count of the posterior probability for every classification and example, is dreamy away with a capacity call calculate posterior in line 10of pseudo code. It will be clarified next.

We utilize Bayes hypothesis to ascertain the posterior probability. When we have our posterior probabilities and the set of conditional probabilities for each one characteristic in every classification, it could be made as the accompanying formula:

$$P\ (category|W) = P\ (category)(w_i|category). \tag{6.12}$$

In the equation, W is the situated of term frequencies of each one statement in the content to be grouped that additionally shows up in the term check from the trainig part. This means we take the posterior probability for the classification and reproduce it with the conditional probability for each one term in that class found in the content.

The point, when this is finished every classification, one of them has been found to have the most astounding posterior probability for that content. The content is then considered having a place with that class. At last, the decided classification is contrasted with the real class of the specimen. Once more, since we are utilizing the training situated, the correct class of each one specimen is known. Assuming that the classifications are one and the same, we build a counter letting us know what number of great arrangements we have made. The point when every confirmation test has been endeavoured classified, we can then evaluate efficiency.

6.4 Performance Evaluation

After the completion of training and testing phase for both supervised and unsupervised machines learning algorithm which are Naïve Bayes and Self Organising Map, the results of both techniques with their prepossessed data are compared.

The evaluation criterion use to measure accuracy of an Intrusion Detection System is its ability to correctly identify connections as normal or attack connections. This criterion of measuring the performance of Intrusion Detection System is called detection rate. There are four possible outcome combinations which can be achieved during evaluating the results which are:

True Positive (TP): Attacks correctly predicted\classified as attacks

False Positive (FP): Normal incorrectly predicted\classified as attack

True Negative (TN): Normal correctly predicted\classified as normal

False Negative (FN): Attacks incorrectly predicted\classified as normal

The *True Positive Rate* (*TPR*) is the proportion of ambushes discovered by the IDS to the amount of actual assaults in the information set. It is also called *Recall* (*R*).

$$R = \frac{TP}{(TP + FN)} \text{ ,} \tag{6.13}$$

where *R* is the Recall, *TP* is number is true positive connections and *FN* is the number of false normal connections.

The *False Positive Rate (FPR)* is the amount of ordinary connections that are misclassified as ambushes partitioned by the amount of typical connections in the information set.

$$FPR = 1 - R = \frac{FN}{(TP + FN)} \text{ ,} \tag{6.14}$$

where *FNR* is the False Positive Rate, *TP* is number is true positive connections and *FN* is the number of false normal connections.

The criterion which is important for evaluation of Network Intrusion Detection is false positive rate because it can highly affect the performance of system as it will predict the normal connections as attack connections and then there will be more filters to apply on those connections which can slow down the system performance. In comparison of our results, it is observed that which of the two machine learning technique gave minimum false positive rate in testing phase.

It is also observed which of the Symbolic Conversion Technique (Indicator Variable or Conditional Probability) has a better effect on the performance of the systems and evaluation of results. Four combinations of results are made by finding the SOM performance with conditional probability symbolic conversion technique and indicator variable symbolic conversion technique. Then the performance for Naïve Bayes is observed both with the conditional probability conversion technique and indicator variable conversion technique.

Finally, the Accuracy *(AC)* of the both techniques is evaluated. The Accuracy (*AC*) is the ratio of total number of correct predictions\classification to the actual data of size. Mathematically, it can be represented by following equation.

$$AC = \frac{TP + TN}{(TP + TN + FP + FN)}\ , \tag{6.15}$$

where *AC* is the overall Accuracy, *TP* is number is true positive connections, *TN* is number is true negative connections, *FP* is number is false positive connection and *FN* is the number of false normal connections.

6.5 Experiment

This section provides in detail the experimental results and the discussion around it. The structure is arranged such that it first explains the environment in which the implementation was performed, followed by the results obtained and finally the evaluation of SOM in particular trained on conditional probability and on indicator variable pre-processed data.

Furthermore, this section will concentrate on the investigations and their outcomes about that were depicted in past section. The tests will be utilized as a verification of idea. The analyses were directed on the intrusion identification dataset called NSL-KDD'99 glass dataset. The information utilized as a part of this study is those proposed in the NSL-KDD'99 for intrusion detection [6]. Which are by and large utilized for benchmarking intrusion detection issues? They set up an environment to gather *TCP/IP* dump columns from a host found on a reproduced military system. Every *TCP/IP* connection is portrayed by 41 discrete and constant characteristics and marked as either normal, or as an attack, with precisely one particular assault type. We assess the execution of our framework by the discovery rate and the false positive rate. The detection rate is the amount of ambushes distinguished by the framework separated by the amount of strike in the information set. The false positive rate is the amount of ordinary connections that are misclassified as strike isolated by the amount of typical connections in the information set.

6.6 Environment

The implementation was completed on a portable computer with 1.60 GHz of dual CPU and 4 GB RAM. Because of the impediment in the accessible memory and preparing force, it was not conceivable to utilize the full dataset portrayed prior. The SOM and Naïve Bayes calculations are implemented in MATLAB R2013A.

6.7 Results

With a specific end goal to break down and think about the execution of the Naïve Bayes and Self Organizing Map algorithms, measurements like the indicator variables and conditional probability transformation procedures were utilized with the end goal of getting outcomes of False Positive and exactness. The training dataset holds 500 connection vectors out of which 250 are ordinary connections and 250 are assault connections. The test dataset additionally holds 500 connection vectors. As MATLAB R2013A is slow, a couple of connection vectors were taken both for preparing and testing. The point when the conditional probability transformation method was utilized, we had 44 qualities of every connection and when indicator variable conversion procedure was utilized, we had 122 characteristics for every connection.

The detection rate is the amount of ambushes discovered by the framework isolated by the amount of assaults in the information set. It is identical to recall. The false positive rate is the amount of ordinary connections that are misclassified as ambushes partitioned by the amount of typical connections in the information set.

The main calculation that was actualized was a Naïve Bayesian that consolidated a multinomial methodology. The classifier was trained and tested on both the test set and training set utilizing Indicator Variables and Conditional Probability conversion procedures.

By utilizing these systems, we got results for false positive rate and exactness (%) as demonstrated in table 6.5 and 6.6. The point when utilizing pointer strategy, the exactness (%) of Naïve Bayes Classifier is 98.4% while utilizing restrictive likelihood precision rate was 54.8%. There is a high False Positive rate in utilizing Conditional Probability as compared to Indicator Variable conversion method.

Actually, inside Naïve Bayesian systems, when a class is displayed by a low number of preparing occurrences, then it prompts a weak learning regards to this class and thus to a misclassification of testing connections truly fitting in with it. Subsequently, we can have new testing examples truly described by properties' qualities which veer off from those describing these two classes in the training set. These cases are not taken in the development stage and their ensuing classes when applying the deduction component are for the most part not right.

A Naïve Bayes Classifier [8] is a basic probabilistic classifier dependent upon applying Bayes' hypothesis with solid (guileless) freedom presumptions. Because of the underlying probability design efficiency be "autonomous marked feature prototype. Conditional ruling the accurate letter of the fair chance type, Naïve Bayes classifiers puissance be prepared proficiently in a supplied seizing in establishing. Disregarding their undesigning map and obviously over-improved concessions, Naïve Bayes classifiers is made up of many network certifiable conditions. A far stretch forting correlation through other arranging techniques demonstrated that Bayes characterization is beaten by additional current methodologies.

Favourable element of the Naïve Bayes classifier is that it just obliges a little measure of preparing information to gauge the parameters fundamental for arrangement.

The Self Organizing Map was decided to have 30 by 30 dimensions. SOM was prepared on dataset holding 500 connection vectors in 15 iterations with a radius of 15, the learning rate was decided to be 0.1 in begin and it decreases with time. The Mexican hat function was utilized as the upgrade rule.

In the wake of testing utilizing SOM and Naïve Bayes Machine Learning methods on dataset pre-processed utilizing Indicator Variables and Conditional Probability conversion approaches, we get the accompanying effects for false positive rate and exactness.

Table 6.5 **- Results for Detecting Attack Connections**

Machine Learning Algorithms	Conversion Techniques	False Positive	Accuracy (%)
Self-Organizing Map	Indicator Variables	14/250	94.4
	Conditional Probability	15/250	94
Naïve Bayes	Indicator Variables	4/250	98.4
	Conditional Probability	113/250	54.8

The table above shows that the best outcomes about are gotten via preparing Naïve Bayes Machine Learning Approach on dataset pre-processed by Indicator Variable Symbolic Conversion strategy. The outcomes got via preparing of Self Organizing Map on both kinds of pre-processed dataset are very nearly the same and good. The most noticeably bad outcomes are gotten in the wake of preparing Naïve Bayes calculation on dataset pre-processed utilizing Conditional Probability Symbolic Conversion approach. We likewise conclude from the outcomes about that Indicator Variable Symbolic Conversion methodology is superior to Conditional Probability Conversion approach.

In the wake of testing, utilizing SOM and Naïve Bayes Machine Learning calculation on dataset pre-processed indicator variables and conditional probability change procedures; we get the accompanying effects for false positive rate and exactness (%).

Table 6.6 - **Results for Detecting Normal Connections**

Machine Learning Algorithms	**Conversion Techniques**	**False Negative**	**Accuracy (%)**
Self-Organizing Map	Indicator Variables	3 /250	98.8
	Conditional Probability	16 /250	93.6
Naïve Bayes	Indicator Variables	198 /250	20.8
	Conditional Probability	130 /250	48

As saw from the after effect of false negative rate, we find that Self Organizing Map method outflanks Naïve Bayes Machine Learning Algorithm. There is very little drawback of false negative rate as it does not influence the security of any connection. It is a fact that false positive rate may influence the execution of the detection systems as additional connections might be there to be examined and checked.

This work might likewise show the maps that are created in the training and testing of Self Organizing guide for both sort of pre-processed dataset. The maps are created in MATLAB R2013A while preparing methodology of SOM. Distinctive shade codes are utilized to separate the prepared ordinary connections and ambush connections. Throughout testing the diverse colours are relegated to false positive and false negative identification of the connections.

A graphical representation of self-organizing map trained and tested on dataset pre-processed by Conditional Probability transformation is shown in the following sections.

6.8 SOM Trained on Conditional Probability Processed Dataset

As dataset for SOM training was chosen to be 30 by 30 dimensions, so when data is preprocessed using Conditional Probability and given to SOM, it organizes itself through SOM technique and scattered on that 30 by 30 dimensions. If connection is normal, it is represented by yellow circle and if it is attack connection, it is represented by red circle. Each yellow and red circle corresponds to one normal and attack input neuron respectively which was presented to SOM. This process is called training process. After graph processed all input neurons, each output neuron is presented to the same graph one by one and the its placement on graph is depicted by using Euclidian distance formula from output neuron to each input neuron and the input neuron which has minimum distance from output neuron is marked for output neuron placement. If the marked neuron is yellow, output neuron would be designated as normal and if red output neuron would be designated attack connection accordingly and then it is determined the presented output\test neuron was actually normal or attack connection. If result given by SOM for output neuron matches with actual neuron’s result then it is not marked or it can be said that it is given same yellow\red color of input neuron to which it mapped. If a connection\neuron presented to map was originally normal connection but SOM prompts it as an attack connection\neuron, it will be marked as blue and it is called false positive connection. In contrast, if a connection\neuron presented to map was originally attack connection and SOM marks that as normal connection then it is called false positive connection and it is marked with black circle in figure 6.3. If SOM will be unable to decide the output of connection\neuron given to it, it is called undecided connection and has been marked as magenta color circles in figure 6.3.

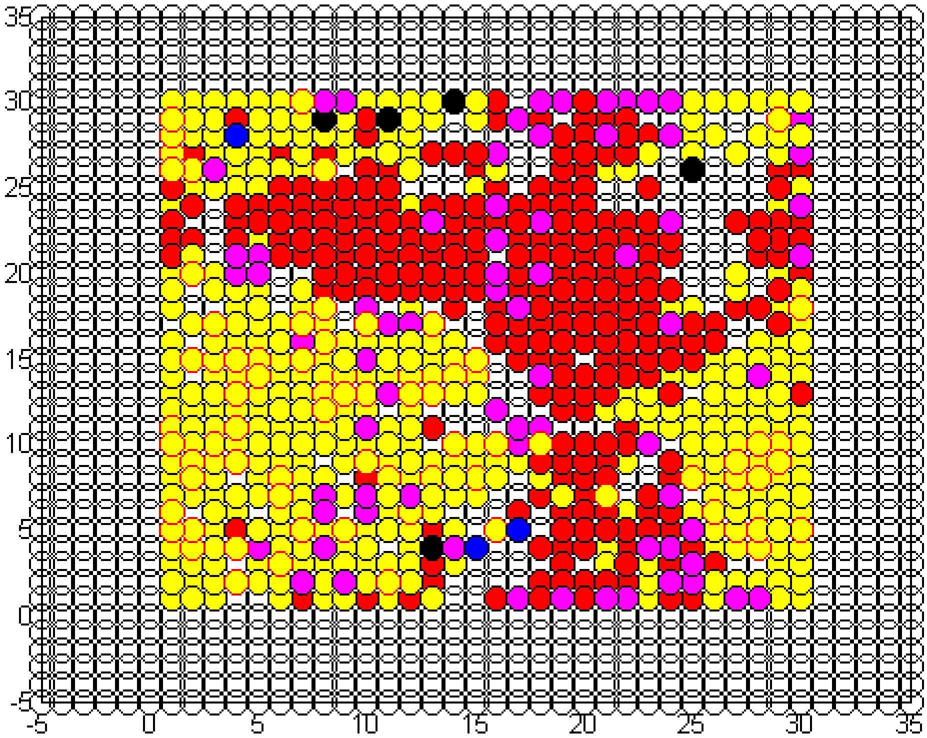


Figure 6.3 **- SOM Prepared with Conditional Probability Transformed Dataset**

Legend

Yellow	Trained Normal Connections
Red	Trained Attack Connections
Blue	False Negative Connections
Black	False Positive Connections
Magenta	Undecided Connections

6.9 SOM Trained on Indicator Variable Processed Dataset

The logic to form the graph shown in figure 6.4 is same as described in previous section. The training and testing process are also same. The only difference is that data presented to graph is pre-processed using a different data conversion technique which is Indicator Variable instead of Conditional Probability.

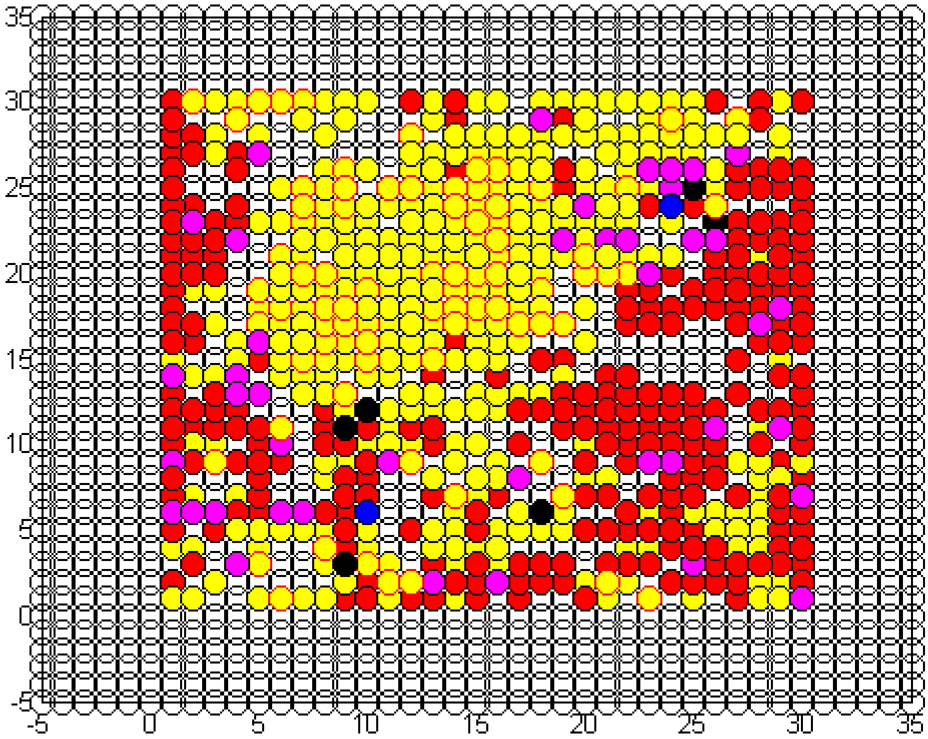


Figure 6.4 **- SOM Prepared with Indicator Variable Transformed Dataset**

Legend

Yellow	Trained Normal Connections
Red	Trained Attack Connections
Blue	False Negative Connections
Black	False Positive Connections
Magenta	Undecided Connections

6.10 Summary

NSL-KDD comprises of the same characteristics as KDD'99. The KDD'99 dataset comprises of 41 characteristics and one class characteristic. This dataset has a twofold class characteristic. The training dataset is chosen from NSL-KDD dataset to incorporate 500 connection vectors out of which 250 are ambush and 250 are ordinary connections. The test dataset additionally incorporates 500 connection vectors. There are numerous ASCII values in the dataset speaking to typical connection with the saying "ordinary" and strike connections with the statement "ambush". NSL-KDD information set has an alternate connection record all connection record has add up to 41 features. 38 of these characteristics are numeric and three of them are typical consequently, we delineate esteemed credits to numeric-esteemed qualities. NSL KDD information set have two sorts of information set changed over to numeric qualities utilizing two separate methodologies which are Conditional Probability and Indicator Variable Symbolic transformation strategy.

The point when the KDD Cup 1999 dataset is pre-processed utilizing Indicator Variables transformation, the 41 characteristics of every connection vectors are expanded to 122 characteristics as three typical traits which are Protocol (3 attributes), Service (70 attributes) and Flag (11 attributes) are displaced by their indicator variables vector each.

The second technique is to change over a typical characteristic with a cluster of Conditional Probability for getting each one class given the quality that has specific typical worth. The Conditional Probability Conversion methodology is connected to typical characteristics of KDD'99 dataset. These characteristics incorporate Protocol, Service and flag characteristic.

Both supervised and unsupervised machine learning methods are prepared on the pre-processed preparing dataset. The Naïve Bayes calculation is utilized as supervised machine learning procedure and SOM is utilized as unsupervised machine learning method.

Learning utilizing SOM obliges introduction of SOM as codebook of neurons, preparing of SOM as stated by the calculation talked about above and at long last the test stage. The point when conditional probability change strategy is connected, an information example has 44 sizes; henceforth codebook neurons likewise have 44 measurements. Essentially, when Indicator Variable transformation method is applied, an information example has 122 measurements, thus codebook neurons likewise have 122 sizes. To get the bases of normal characteristics of every neuron through system weight layers. It likewise got normal weights of each one characteristic through system weight layers. In the wake of getting yield of preparing SOM, reproduce the testing or unseen yield information is put on prepared system.

Naïve Bayes Classifier is likewise prepared on 500 connection vectors which are grouped into two classes Normal and Attack. The training calculation is prepared on two kind of pre-processed dataset. The span of this vector is twofold the properties of a characteristic. The vectors of conditional probability of characteristics that were archived in training period of Naïve Bayes are utilized as a part of testing stage.

If the probability of normal connection is more stupendous than attack connection, the information is considered normal else attack. Four mixtures of effects are made by discovering the SOM execution with Conditional Probability and Indicator Variable conversion.

This implementation first included the development of a 4-step process of collecting data, pre-processing it, training and testing it and finally evaluating the algorithm performances. This approach is essential in order to evaluate and compare the effect of Naïve-Bayes as well as SOM algorithms and their respective combinations on the data to be detected for intrusion.

Machine learning procedures have gained impressive consideration around the intrusion detection researchers to address the shortcomings of knowledge base detection systems.

As compare to Knowledge based systems, Machine learning is a framework equipped for obtaining and coordinating the information automatically. The ability of the frameworks to gain for a fact, training, scientific perception, and different means, brings about a framework that can persistently self-improve and along these lines show productivity and adequacy. A machine learning framework generally begins with some information and a corresponding knowledge organization so it can translate, examine, and test the learning gained. Machine learning techniques are based on securing an unequivocal or implied model that empowers the patterns investigated to be sorted.

CHAPTER 7
CONCLUSION

This chapter outlines the conclusion of the implementation and finding of the research carried out. This is followed by the future work that can be potentially taken up to further broaden the research area of this thesis.

This thesis exhibits the usage of a network intrusion detection system utilizing Machine Learning techniques with different conversion methods for the dataset. The work indicates the noteworthy impacts of distinctive conversion applied on the dataset. As Self Organizing Map is unsupervised learning methods, it should perform superior to the Naïve Bayes as it is supervised learning procedure yet Naïve Bayes beats numerous neural system calculations.

The impact of Indicator Variable Conversion and Conditional Probability Conversion is very nearly the same if there should be an occurrence of recognizing attacks in SOM learning yet it differs in the event of catching attacks in Naïve Bayes learning. The Conditional Probability transformation builds the measurement of typical characteristics to two, as every typical characteristic is swapped by two properties in light of the fact that there are two classes Normal and Attack. While measurements expanded by Indicator variables is much substantial as it speaks to each one property of a characteristic by aggregate number of properties in a characteristic, for instance if a characteristic has 70 attributes, each one attribute might be swapped by 70 attributes. Keeping in mind the end goal to be viable, false positives must be decreased. Of these sorts of slips, the false negatives are more hazardous for IDS, on the grounds that it does not recognize a caution as an occurrence, and gives it a chance to pass through the sifting technique.

In view of the trials completed in the dissertation and their comparing effects, we can state the accompanying:

- Machine learning is a powerful approach which could be utilized within the field of Computer Security.

- The innate nature of Machine Learning calculations makes them more suited to the intrusion detection field of data security. On the other hand, it is not restricted to intrusion detection. The creators in [119] have created an instrument utilizing machine figuring out how to induce access control approaches where approach solicitations and reactions are produced by utilizing learning calculations. These are viable with new arrangement particular dialects like XACML [119]. Also, a classifier-based methodology to relegating clients to parts and bad habit versa is depicted in [120].

- It is conceivable to break down immense amounts of review information by utilizing Machine Learning procedures, which is generally an amazingly troublesome errand.

The accompanying areas recommend a few suggestions for future work mention a real world application of machine learning to information security and also discuss on the security of machine learning.

7.1 Contribution of Thesis

As presented above, there are three components to this research work, which correspondingly represent its contributions.

In the first place, recognizing disparities in the findings is reported in the inscription. This has prompted to an observational examination of the KDD Cup '99 information set, which has revealed a few underlying, reasons for the disparities. This research work implements interpretation of this dataset using Symbolic Conversion, Indicator Variable and Conditional Probability data processing techniques, depending on the data attributes.

Secondly, this work implements two distinctive machine learning methodologies, both supervised and unsupervised, which are Naïve Bayes and Self Organising Maps, and focus on the relative qualities and their performance with a dataset using different combinations of their pre-processing techniques.

Thirdly, this work gives an assessment of the execution of these calculations that may permit somebody who wishes to utilize one of these methodologies to see how precise the methodology is and under what data pre-processing technique it works well.

This exploration exhibits a rationale as to why Naïve Bayes and Self Organizing Map machine learning approaches are more equipped for infectious intrusions.

In this approach, keeping in mind the conclusion is to understand the statistical ability of machine figuring out how to the field of PC security, it is fundamental to explore different avenues regarding different Machine Learning plans towards tending to security-related issues and pick the particular case that is the most suitable to the issue nearby.

7.2 Future Work

Machine Learning is an experimental science. A learning system which may be suited to a specific issue may not so much perform well at another issue. Additionally, a learning technique may have numerous configurable parameters, which may bring about an alternate execution.

In view of the above realities, the future work to this thesis can be summarized as follows:

1. In this theory, two learning calculations were tried and looked at. The Weka Machine Learning toolbox offers a gathering of numerous other learning plans, which could be tried and assessed. Also, it may be conceivable to further enhance the execution of the strategies utilized within this thesis towards intrusion detection by streamlining these parameters.

2. Owing to the restricted preparing force, memory accessible for the experiments directed and the extent of the postulation, a lessened subset of the genuine dataset was utilized. These examinations might be rehashed by taking the whole dataset which may further enhance the execution of the learner.

3. Hence, indicator variables to a great extent build the sizes of dataset. Matlab was utilized for execution; subsequently the transforming was moderate. Because of this reason the training and testing was performed on a little dataset. In future C language might be utilized for quick handling and for usage of continuous Intrusion detection systems.

4. Another approach to further enhance the execution of the machine learners might be to prepare them with a higher degree of negative occurrences. Since it was not inside the extent of this examination to discover the ideal "positive-negative-proportion", it could improve the execution of the machine learners. So the third proposal might be excessively assess this degree.

References

[1]. E. Hernandaz-Pereira, J.A. Suárez-Romero, O. Fontenla-Romero and A. Alonso-Betanzos, "*Conversion Methods for Symbolic Features: A Comparison Applied to Intrusion Detection*". Expert Systems with Applications: An International Journal Volume 36, Issue 7, Pages 10612-10617, September, 2009

[2]. Saroj Kumar Panigrahy, Jyoti Ranj and Mahapatra, Jignyanshu Mohanty, and Sanjay Kumar Jena, "*Anomaly Detection in Ethernet Networks using Self Organizing Maps*". National Institute of Technology Rourkela, Page 769 - 800, Odisha, India.

[3]. Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu and Ali A. Ghorbani, "*A Detailed Analysis of the KDD CUP 99 Dataset*". CISDA'09 Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications, IEEE Press Piscataway, NJ, Pages 53-58, USA, 2009

[4]. H. Gunes Kayacik, A. NurZincir-Heywood and Malcolm I. Heywood, "*A Hierarchical SOM based Intrusion Detection System*". Journal of Engineering Applications of Artificial Intelligence, Volume 20, Issue 4, Pages 439-451, June, 2007,

[5]. H. Gunes Kayacik, A. NurZincir-Heywood, Malcolm I. Heywood, "*Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD'99 Intrusion Detection Datasets*". Dalhousie University, Faculty of Computer Science, 6050 University Avenue, Halifax, Nova Scotia. B3H 1W5.

[6]. Mrutyunjaya Panda and Manas Ranjan Patra, "*Network Intrusion Detection Using Naïve Bayes*". IJCSNS International Journal of Computer Science and Network Security, Volume 7, Number 12, December 2007.

[7]. M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "*A Detailed Analysis of the KDD CUP 99 Data Set*". Second IEEE Conference on Computational Intelligence for Security and Defence Applications (CISDA), 2009.

[8]. T. Kohonen, "*Self Organizing Maps*", Third Extended Edition, Springer, Berlin, 2001.

[9]. J. McHugh, "*Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 Dapra Intrusion Detection System Evaluations As Performed By Lincoln*

Laboratory". ACM Transactions on Information and System Security, Volume 3, Number. 4, Page 262–294, 2000.

[10]. Ethem Alpaydin, "*Introduction to Machine Learning*". Second Edition, The MIT Press, Cambridge Massachsetts, London. England, 2010.

[11]. Tom Mitchell, "Machine Learning". Edition 1, McGraw-Hill Science/Engineering/Math; March 1, 1997.

[12]. M. Kumar Sabhnani and G. Serpen, "*Application of Machine Learning Algorithm to KDD Intrusion Detection Dataset within Misuse Detection Context*". Proceedings of IEEE Conference on MLMTA, Page 209-215, 2003.

[13]. Hayoung Oh and Kijoon Chae, "*Real-time Intrusion Detection System based on Self-Organized Maps and Feature Correlations*". ICCIT '08 Proceedings of the Third International Conference on Convergence and Hybrid Information Technology, Volume 2, Pages 1154-1158, 2008.

[14]. Simson Garfinkel and Gene Spafford, "*Practical Unix Security*". Oareilly and Associates, Sebastopol, California, 1991.

[15]. Rick Lehtinen, Deborah Russell, and G.T. Gangemi, "*Computer Security Basics*". Oareilly Media, Inc., California, December 2006.

[16]. Richard Power, "*Current and Future Danger*". A CSI Primer on Computer Crime and Information Warfare, Computer Security Institute, San Francisco, California, 1995.

[17]. E. H. Spafford, "*Crisis and Aftermath*". Communications of the ACM, Volume 32, Number 6, Page 678-687, June 1989.

[18]. Fred Cohen, "*Computer Viruses - Theory and Experiments*". Computers and Security 6, Page 22-35, 1987.

[19]. John F. Shoch and Jon A. Hupp, "*The "worm" programs - early experience with a distributed computation*". A Journal on Communications of The ACM - CACM , Volume 25, Number 3, Page 172-180, 1982.

[20]. Ken Thompson, "*Reflections on Trusting Trust*". Communications of the ACM, Volume 1, Number 3, Page 21-31, July 1987.

[21]. B. W. Lampson, "*Protection*". In Proceedings of the Fifth Annual Princeton Conference on Information Science Systems, Pages 437-443, 1971. Reprinted in Operating System Review, Volume 8, Number 1, Pages 18-24, January 1974.

[22]. Dorothy E. Robling Denning, "*Cryptography and Data Security*". Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1982.

[23]. D. E. Bell and L. J. La Padula, "*Secure Computer Systems: Mathematical Foundations and Model*". Technical Report M74-244, The MITRE Corporation, Bedford, Massachusetts, May 1973.

[24]. K. J. Biba, "*Integrity Constraints for Secure Computer Systems*". Technical Report ESD-TR-76-372, USAF Electronic Systems Division, Bedford, Massachusetts, April 1977.

[25]. Department of Defence Standard, "*Department of Defence Trusted Computer System Evaluation Criteria*". DOD 5200.28-STD. U.S. Government Printing Office, December 1985.

[26]. R. Heady, G. Luger, A. Maccabe, and M. Servilla, "*The Architecture of a Network Level Intrusion Detection System*". Technical Report, Department of Computer Science, University of New Mexico, August 1990.

[27]. J. P. Anderson, "*Computer Security Threat Monitoring and Surveillance*", Technical Report, Fort Washington, Pennsylvania, April 1980.

[28]. Staniford-Chen, S., Tung, B., Porras, P., Kahn, C., Schnackenberg, D., Feiertag, R., and Stillman, M."*The Common Intrusion Detection Framework - Data Formats*". 1998.

[29]. Siraj A., Vaughn R., and Bridges S., "*Intrusion Sensor Data Fusion in an Intelligent Intrusion Detection System Architecture*". In Proceedings of the 37th Hawaii International Conference on System Sciences, 2004

[30]. D. E. Denning, "*An Intrusion-Detection Model*". Proceedings of the IEEE Symposium on Security and Privacy, 1986.

[31]. Dorothy E. Denning, "*An Intrusion-Detection Model*". IEEE Transactions on Software Engineering - Special Issue on Computer Security and Privacy, Volume 13, Issue 2, Page 222-232, IEEE Press Piscataway NJ, USA, 1987.

[32]. Smaha, S.E., "*Haystack: An Intrusion Detection System*", IEEE Aerospace Computer Security Applications Conference Fourth Page, Page 37 – 44, Orlando, FL1988.

[33]. T. F. Lunt, R. R. Schell, W. R. Shockley, M. Heckman, and D. Warren, "*A Near-Term Design for the Seaview Multilevel Database System*". In Proceedings of the IEEE Symposium on Security and Privacy, April 1988.

[34]. L. Todd Heberlein, Gihan V. Dias, Karl N. Levitt, Biswanath Mukherjee, Jeff Wood, and David Wolber, "*A Network Security Monito*r". In Proceedings of the IEEE Symposium on Research in Security and Privacy, pages 296-304, May 1990.

[35]. Marcus J. Ranum and Marcus J. Ranum, "*Strategic Security for IP Networks*". Information Section Technical Report, Volume 2, Number 3, Page 46-52 , 1990.

[36]. Teng Chen and Lu, "*Research in Security and Privacy*". Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy, Catalogue number 90CH2884-5, Pages 278 – 284, 1990.

[37]. Mukherjee, B., Heberlein, L. T., and Levitt, K. N., "Network Intrusion Detection," IEEE Network 8(3) pp. 26-41 (1994).

[38]. Anderson, D., Lunt, T. F., Javitz, H., Tamaru, A., and Valdes, A. "*Detecting Unusual Program Behaviour Using the Statistical Component of the Next-Generation Intrusion Detection Expert System*". Technical Report SRI-CSL-95-06, SRI International, 1995.

[39]. Sundaram A., "*An Introduction to Intrusion Detection". Crossroads*-Special Issue on Computer Security, Volume 2, Issue 4, Pages 3-7, March 1996.

[40]. Axelsson, S., "*The Base-Rate Fallacy and its Implications for the Difficulty of Intrusion Detection"*, CCS '99 Proceedings of the 6th ACM conference on Computer and Communications Security, Page 1 - 7, 1998.

[41]. Lane and Brodley, "*Temporal Sequence Learning and Data Reduction for Anomaly Detection*". ACM Transactions on Information and System Security Volume 2, Issue 3, Pages 295-331, August 1999.

[42]. Lee W., Stolfo S. and Mok K., "*A Data Mining Framework Building for Detecting Intrusion Detection Models*." 1999.

[43]. Allen, J., Christie, A. and Stoner, "*State of the Practice of Intrusion Detection Technologies*". Networked Systems Survivability Program, January, 2000.

[44]. Debar, H., Dacier. M. and Wespi, "*A Revised Taxonomy for Intrusion Detection System*". Annals of Telecommunications, Volume 55, Page 361-378, 2000.

[45]. Alessandri, D, Dominique Alessandri, "*Using Rule-Based Activity Descriptions to Evaluate Intrusion-Detection Systems*". Recent Advances in Intrusion Detection, Fourth International Workshop, RAID2001, Davis, CA, 2001.

[46]. Axelsson, S, "*Intrusion Detection Systems*". A Taxonomy and Survey, Technical Report No 99-15, Department of Computer Engineering, Chalmers University of Technology, Sweden, March 2000.

[47]. Bai, Y., And Kobayashi, H., "*Intrusion Detections: Technology and Development*", Proceedings of the 17th International Conference on Advanced Information Networking and Applications, Page 710-717, March 27-29, 2003.

[48]. Yu Chen, Yu-Kwong Kwok, and Kai Hwang, "*Trusted Grid Computing with Security Binding and Self-defense against Network Worms and DoS Attacks*", International Workshop on Grid Computing Security and Resource Management in conjunction with the ICCS-2005, 2005.

[49]. Phillip J. Brooke, Richard F. Paige, and Jeremy L. Jacob, "*A CSP Model of Eiffel's SCOOP*". Formal Aspects of Computing, Volume 19, Number.4, 2007.

[50]. Jeyanthi Hall, "*Detection of rogue devices in wireless networks"*, Carleton University Ottawa, Ont., Canada 2007

[51]. Eduardo Mosqueira-Rey, "*A Misuse Detection Agent for Intrusion Detection in a Multi-agent Architecture*". KES-AMSTA '07 Proceedings of the 1st KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, Pages 466-475, 2007.

[52]. Magnus Almgren, Ulf Lindqvist and ErlandJonsson, "*A Multi-Sensor Model to Improve Automated Attack Detection*". Proceeding RAID '08 Proceedings of the 11th International Symposium on Recent Advances in Intrusion Detection, Pages 291 – 310, Springer-Verlag Berlin, Heidelberg, 2008.

[53]. Naeimeh Laleh and Mohammad Abdollahi Azgomi, "*A Taxonomy of Frauds and Fraud Detection Techniques*". ICISTM, Communications in Computer and Information Science, Volume 31, Pages 256-267, Springer, 2009.

[54]. Yong (Yates) Lin, Zhengyi Le, Eric Becker and Fillia Makedon, "*Acoustical Implicit Communication in Human-Robot Interaction*", PETRA '10 Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments, Article No. 5, ACM New York, USA, 2010.

[55]. P fleeger, C. and P fleeger,S."*Security in Computing*". Prentice Hall. 2003.

[56]. Jackson, T., Levine, J., Grizzard, J., and Owen, H, "*An Investigation of a Compromised Host on a Honeynet being Used to Increase the Security of a Large*

Enterprise Network". In Proceedings of the 2004 IEEE Workshop on Information Assurance and Security, 2004.

[57]. Proctor, P, "*The Practical Intrusion Detection Handbook*". Prentice Hall, 2001.

[58]. Stillerman, M., Marceau, C., and Stillman, M. "*IntrusionDetection for Distributed Applications*". Communications of the ACM, Volume 42, Page 62-69, 1999.

[59]. Botha, M. and Von Solms R., "*Utilizing Fuzzy Logic and Trend Analysis for Effective Intrusion Detection*". Computers and Security, Volume 22, Page 423-434, 2003.

[60]. Alex Lukatsky, "*Protect your Information with Intrusion Detection*". BPB Publications, May 2004.

[61]. Comer D., "Computer Networks and Internets with Internet Applications", Fourth Edition, Pearson Education, 2003.

[62]. Stefan Axelsson, "*Research in Intrusion Detection Systems: A Survey. Technical Report*". Department of Computer Engineering, Chalmers University of Technology, Goteborg, Sweden, December, 1998.

[63]. P. Helman and G. Liepins, "*Statistical Foundations of Audit Trail Analysis for the Detection of Computer Misuse*". IEEE Transactions on Software Engineering, Volume 19, Issue 9, Page 886-901, September 1993.

[64]. R. Heady, G. Luger, A. Maccabe, and M. Servilla, "*The Architecture of a Network Level Intrusion Detection System*". Technical Report, Department of Computer Science, University of New Mexico, August 1990.

[65]. K. Chen, "*An Inductive Engine for the Acquisition of Temporal Knowledge*". Department of Computer Science, University of Illinois at Urbana-Champaign, 1988.

[66]. Henry S. Teng, Kaihu Chen, and Stephen C Lu, "*Security Audit Trail Analysis using Inductively Generated Predictive Rules*". In Proceedings of the Sixth IEEE Conference on Artificial Intelligence Applications, Piscataway, New Jersey, March 1990.

[67]. Kevin L. Fox, Ronda R. Henning, Jonathan H. Reed, and Richard Si-Monian, "*A Neural Network Approach towards Intrusion Detection*". In Proceedings of the 13th National Computer Security Conference, Pages 125-134, Washington, DC, October 1990.

[68]. Patrick Henry Winston, "*Artificial Intelligence*". Addison Wesley Publishing,

Massachusetts, Third edition, Page 411-422, 1992.

[69]. Peter Cheeseman, Robin Hanson, and John Stutz, "*Bayesian Classification with Correlation and Inheritance*". 12th IJCAI International Joint Conference on Artificial Intelligence (IJCAI), Volume 2, August 1991.

[70]. Peter Cheeseman and John Stutz, "*A Bayesian Classification System (Auto Class) – Theory and Experiments".* Proceedings of the 5th International Conference on Machine Learning, Pages 54-64, June 1988.

[71]. Judea Pearl, "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference". Morgan Kaufmann Publishers Inc., San Mateo, California, 1988.

[72]. Eugene Charniak, "*Bayesian Networks Without Tears*". Al Magazine, Volume 12, Issue 4, Pages 50-63, 1991.

[73]. Shiuhpyng Winston Shieh and Virgil D. Gligor, "*A Pattern Oriented Intrusion Model and its Applications*". Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, Page 327-342, May 1991.

[74]. Steven R. Snapp and Stephen E. Smaha, "*Signature Analysis Model Definition and Formalism*". Proceedings of the Fourth Workshop on Computer Security Incident Handling, Denver, Colorado, August 1992.

[75]. Joseph C. Giarratano, "*Clips Version 5.1 User's Guide*". NASA, Lyndon B. Johnson Space Centre Information Systems Directorate, Software Technology Branch, March 1992.

[76]. Charles L. Forgy, "*RETE: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem*". Artificial Intelligence, Volume 19, 1982.

[77]. Phillip A. Porras and Richard A., "*Kemmerer. Penetration State Transition Analysis - A Rule-Based Intrusion Detection Approach*". 8th Annual Computer Security Applications Conference, Pages 220-229, IEEE Computer Society Press, November 30 - December 4, 1992.

[78]. Morris I. Bolsky and David G. Korn, "*The KornShell Command and Programming Language*". Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[79]. T. D. Garvey and T. F. Lunt, "*Model based Intrusion Detection*". Proceedings of the 11th National Computer Security Conference, Page 372-385, October 1991.

[80]. Mitchell T., "*Does Machine Learning Really Work?".* AI Magazine, Page 11-20, 1997.

[81]. Hall M. and Smith L., "*Practical Feature Subset Selection for Machine Learning*".

Proceedings of the Australian Computer Science Conference (University of Western Australia), University of Waikato, 1996.

[82]. Nilsson N. J., "*Introduction to Machine Learning*". An Early Draft of a Proposed Textbook, 1996.

[83]. Pietraszek T., "*Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection*". Recent Advances in Intrusion Detection, Volume 3224/2004, Page 102-124, 2004

[84]. Hendrickx I. and Van Den Bosch, "*A* Hybrid Algorithms with Instance Based Classification". Proceedings of the 16th European Conference on Machine Learning, Pages 158-169, 2005.

[85]. Ghahramani Z. "Unsupervised Learning". Page 72-112. Berlin: Springer-Verlag. 2004.

[86]. Segaran T." *Programming Collective Intelligence*". O'Reilly Media, Inc., 2007.

[87]. Rasmussen C. E., and Williams C. K. I., "*Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*". The MIT Press, December 2005.

[88]. D. Barman, K. Claffy, M. Faloutsos, M. Fomenkov, H. Kim and K. Lee, "*Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices*". ACM SIGCOMM Conference on Emerging Networking Experiments and Technologies (CoNEXT), Article 11, New York, NY, USA, 2008.

[89]. Bao, Y., Lu Y., and Zhang J., "*Forecasting Stock Price by SVMs Regression*". AIMSA, Volume 3192, Page 295-303, 2004.

[90]. Hastie, T., Tibshirani, R., and Friedman, J. "*The Elements of Statistical Learning*". Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2001.

[91]. P. Domingos and M. J. Pazzani, "*On the Optimality of the Simple Bayesian Classifier under Zero-One Loss*". Machine Learning, Volume 29, Number 2-3, Page 103-130, 1997.

[92]. Fisher, R. A, "*The Use of Multiple Measurements in Taxonomic Problems*". Annals of Eugenics, Journal 7, Number 2, Page 179-188, 1936.

[93]. Draper, N. R., and Smith, H, "*Applied Regression Analysis*". John Wiley and Sons, Inc., New York, 1981

[94]. Mika, S., Ratsch, G., Weston, J., Scholkopf, B., and Mullers, K. R, "*Fisher Discriminant Analysis with Kernels*". Proceedings of the 1999 IEEE Signal

Processing Society Workshop Neural Networks for Signal Processing IX, Page 41-48, 1999.

[95]. Box G. E. P, "*A General Distribution Theory for a Class of Likelihood Criteria*". Biometrika, Journal 36, Number 3, Page 317-346, December 1949.

[96]. Siotani, M., Hayakawa, T., and Fujikoshi Y., "*Modern Multivariate Statistical Analysis: A Graduate Course and Handbook*". American Sciences Press, Columbus, OH, 1985.

[97]. Seymour G., "*A Predictive Approach to the Random Effect Model*". Biometrika**,** Journal 61, Number 1, Page 101-107, 1974.

[98]. Verbyla, D. L., and Litvaitis, J. A., "*Resampling Methods for Evaluating Classification Accuracy of Wildlife Habitat Models*". Environmental Management, Journal 13, Number 6, Page 783-787, 1989.

[99]. S. J. Russell and P. Norvig, "*Artificial Intelligence: A Modern Approach (International Edition)*". Pearson US Imports & PHIPEs, November 2002.

[100]. V. V. et al, "*Automation and Remote Control - Pattern Recognition using Generalized Portrait Method*", Telemekhanika, Volume 24, Number 6, Page 774-780, Moscow, June 1963.

[101]. B. et al, "*A Training Algorithm for Optimal Margin Classifiers*", Proceedings of the 5th Annual Workshop on Computational Learning Theory COLT'92, Page 144-152. ACM Press, 1 ed., 1992.

[102]. K.-B. D. et al, "*Multiple Classifier Systems: Which is the Best Multiclass SVM Method? An Empirical Study*". 6th International Workshop on Multiple Classifier System, Page 278-285, 2005.

[103]. A.D. Gordon, "*Classification*". 2nd Edition. Chapman and Hall/CRC Press, Page 117-178, June 1999.

[104]. Steinbach, M., G. Karypis, and V. Kumar, "*A Comparison of Document Clustering Techniques*", KDD Workshop on Text Mining, Page 109-111, Boston, MA, August 20, 1999.

[105]. Hartigan, J., "*Clustering Algorithms*". John Wiley and Sons, New York, 1975.

[106]. Berkhin, P., "*Survey of Clustering Data Mining Techniques*", Technical Report, Accrue Software, San Jose, CA, 2002.

[107]. Forgy, E., "*Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification*", Biometrics, Volume 21, Issue 1, Page 768-780, 1965.

[108]. Kaufman, L. and Rousseeuw, P.J., "*Finding Groups in Data: An Introduction to Cluster Analysis",* Wiley-Interscience (Series in Applied Probability and Statistics), ISBN 0-471-87876-6, New York, 1990

[109]. T. Kohonen, "Self-Organizing Maps", Springer Series in Information Sciences, Volume 30, 3rd Edition, Springer, Berlin, Heidelberg, New York, 2002.

[110]. S. Kaski, J. Kangas, and T. Kohonen, "*S*elf-organizing Map (SOM)". Neural Computing Surveys, 1st Edition, Page 102-350, 1998.

[111]. J. Vesanto, "*Neural Network Tool for Data Mining: SOM Toolbox*". 5th International Symposium on Tool Environments and Development Methods for Intelligent Systems (TOOLMET2000), Pages 184-196, Oulu, Finland, April 13-14, 2000.

[112]. J. Vesanto, "*Data Exploration Process Based on the SOM*'.
Helsinki University of Technology, 2002.

[113]. J. Moody, C. J. Darken, "*Fast Learning Networks of Locally- Tuned Processing Units*". Neural Computation, Volume 1, Number 2, Page 281-294, 1989.

[114]. J. Himberg, J. Ahola, E. Alhoniemi, J. Vesanto, and O. Simula, "*The Self Organizing Map as a Tool in Knowledge Engineering*". Pattern Recognition in Soft Computing Paradigm, Pages 3865, World Scientific 2001.

[115]. Cieslak, D.A. Chawla and N.V. Striegel A., "*Combating Imbalance in Network Intrusion Datasets*". IEEE International Conference on Granular Computing, Volume 10, Number 12, Page 732 – 737, 2006.

[116]. Kendall Kristopher., "*A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems*". Cambridge MIT, 1999.

[117]. Shyu, M.-L., Sarinnapakorn, K., Kuruppu-Appuhamilage, I., Chen, S.-C., Chang, L. and Goldring, T. "*Handling Nominal Features in Anomaly Intrusion Detection Problems*". 15th International Workshop on Research Issues in Data Engineering - Stream Data Mining and Applications, 2005.

[118]. Hernández-Pereira, E., Suárez-Romero, J. A., Fontenla-Romero, O. and Alonso-Betanzos, A., "*Conversion Methods for Symbolic Features: A Comparison Applied to an Intrusion Detection Problem*". Expert Systems with Applications, 2009.

[119]. Evan Martin and Tao Xie, "*Inferring Access-Control Policy Properties via Machine Learning*". Seventh IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'06), Policy, Page 235-238, Washington, DC,

2006,

[120]. Shengli Sheng and Sylvia L. Osborn, "*A Classifier-based Approach to User-role Assignment for Web Applications*". Secure Data Management of Computer Science, Volume 3178, Page. 163-171, 2004.

[121]. M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, *"A Detailed Analysis of the KDD CUP 99 Data Set, 2009*" IEEE Int. Conf. Comput. Intell. Security Defense Appl., 2009, Page. 53-58