“Data mining framework for Computer Network Security Management - A study with special reference to IT industrial units in Pune region during 2012-2014”

A Thesis submitted to
Tilak Maharashtra Vidyapeeth
Pune

For the Degree of

Doctor of Philosophy (Ph. D.)

Under The Faculty of Management

By
Mrs. Neelam S. Chandolikar
B.Sc. (Comp), MCA, M.Phil

Under the Guidance of
Dr. Vilas D. Nandavadekar
B.Sc, MCA, MPM, Ph.D.

July 2014

DECLARATION

I hereby declare that the thesis entitled “Data mining framework for Computer Network Security Management - A study with special reference to IT industrial units in Pune region during 2012-2014” completed and written by me has not previously formed the basis for the award of any Degree or other similar title upon me of this or any other Vidyapeeth or examining body.

Date :

Place:

Mrs. Neelam S. Chandolikar

Research student

C E R T I F I C A T E

This is to certify that the thesis entitled "**Data mining framework for Computer Network Security Management - A study with special reference to IT industrial units in Pune region during 2012-2014**" which is being submitted herewith for the award of the Degree of Vidyavachaspati (Ph.D.) in Computer Managemnt of Tilak Maharashtra Vidyapeeth, Pune is the result of original research work completed by Smt. NEELAM SUDHIR CHANDOLIKAR under my supervision and guidance. To the best of my knowledge and belief, the work incorporated in this thesis has not formed the basis for the award of any Degree or similar title of this or any other University or examining body upon her.

Dr. Vilas D. Nandavadekar

(Research Guide)

Place :

Date :

ACKNOWLEDGEMENT

Completion of this doctoral dissertation was possible with the support of several people. I would like to express my sincere gratitude to all of them.

I am extremely grateful to my research guide, Dr. Vilas D. Nandavadekar, Director, Sinhgad Institute of Management, Pune for his valuable guidance, scholarly inputs and consistent encouragement I received throughout the research work. A person with great vision and positive disposition, Sir has always made himself available to clarify my doubts despite his busy schedules and I consider it as a great opportunity to do my doctoral programme under his guidance and to learn from his research expertise. Thank you Sir, for all your help and support.

I thank Dr. R. M. Jalnekar ,Director, Vishwakarma Institute of Technology for motivating staff for research work . I sincerely thank Prof. M. L. Dhore for his valuable suggestions, guidance and concise comments on my research work time to time and Prof. A. M. Kulkarni for his consistent support. Some faculty members of the Institute have been very kind enough to extend their help at various phases of this research, whenever I approached them, and I do hereby acknowledge all of them.

I thank Hon. H. K. Abhyankar for motivation and Dr. R. N Dhamdhere for giving direction to thesis work.

Finally, I thank my Aai for her unconditional support. I am very much indebted to my family, my husband Mr. Sudhir Chandolikar, my children Suhani, Suyash and my parents who supported me in every possible way to see the completion of this work.

Above all, I owe it all to Almighty God for granting me the wisdom, health and strength to undertake this research task and enabling me to its completion.

Mrs. Neelam S. Chandolikar

Abstract

Computer network security is becoming most essential because of tremendous growth of Internet. Interconnected world causes more opportunities for attacker to attack remote computer. Hence, pervasive issue is to make computer network secure. The security of a computer system is in threat because of many reasons .One of the crucial reason is intrusion attack. An intrusion attack causes threat to privacy, reliability and availability of resources. Intrusion Detection process discovers or detects the presence of intrusion attacks. It refers to all processes used in discovering unauthorized uses of network or computer devices.

Generally, an intrusion would cause unauthorized use of resources and challenges network security management. Therefore, effective network security management plays most important role in this interconnected world. Network security management involves various activities like maintaining authorized access, maintaining integrity and reliability of operations. An effective network security management requires identifying threats and then choosing the most effective set of tools to combat them. It comprises various tools like firewall, antivirus, intrusion detection system. Specifically Intrusion Detection System is software designed to detected unusual or abnormal activity. Intrusion detection systems are based on network traffic analysis and their goal is to detect attack in preferably real time.

This study is related to network issues with special reference to IT industrial units in Pune. Being IT hub, many IT companies are situated in Pune region. Computer network security is one of most essential need of these companies. Pune cities IT companies are challenged to extend security to protect a variety of potential vulnerabilities, including Internet connections, communication channels between remote and corporate offices and links between trusted business partners. Unfortunately, the preventive measures employed to secure corporate resources and internal traffic don’t provide the breadth or depth of analysis needed to identify attempted attacks or uncover potential threats across the organization.

Hence, this research specifically deals with computer network security by identifying network security issues in Pune IT industrial units and developing a new method for intrusion detection. Network security issues are identified by survey method.

This research is experimental research. Various experiments are performed using data mining techniques to find best method suitable for intrusion detection purpose. This research investigates data mining techniques .Framework is developed using data mining techniques. This framework is intended to solve network security issue effectively.

INDEX

List of Tables

List of Charts

List of Figures

List of Abbreviations

AI	: Artificial intelligence
ARFF	: Attribute Relation File Format
CSV	: Comma Separated Values
DARPA	: Defense Advanced research Project Agency
DM	: Data Mining
DOS	: Denial of Services
FN	: False Negative
FP	: False Positive
HIDS	: Host Based Intrusion Detection System
HMM	: Hidden Markov Model
ICMP	: Internet Control Message Protocol
IDS	: Intrusion Detection System
IPS	: Intrusion Prevention Systems
IG	: Information Gain
IPS	: Intrusion Prevention Systems
IT	: Information Technology
KD	: Knowledge Discovery
KDD	: Knowledge Discovery in Database
MIT	: Massachusetts Institute of Technology
ML	: Machine learning
NIDS	: Network intrusion detection system
NSL	: Network Simulation Language
OS	: operating system
POD	: Ping of Death
R2L	: Remote to Local

SATAN : Security Administrator Tool for Analyzing Networks

SIDDM : Systematic Intrusion Detection using Data Mining

SVM Support vector machine

TCP : Transmission Control Protocol

TP : True Positive

U2R : User to Remote

WEKA : Waikato Environment for Knowledge Analysis

Chapter 1

Introduction

1.1. Introduction

Computer network security is becoming mandatory for computer networks. Without adequate security use of computer network is risky. Most of the Business and user are reliant on computer network therefore; one cannot take chance to compromise with security. Computer network is continuously evolving. Computer security threats are increasing with pertinent facilities use of computer network.

Though multiple security tools and mechanisms are available but still there is need to find out methods for better performance. These tools and mechanisms are antivirus, firewall, intrusion detection system, security policy etc. intrusion detection systems (IDS) plays significant role in computer network security whereas with increasing opportunities to attacker; IDS need to have reliable and better performance.

1.2. Network security

Network security [8] [9] [10] refers to any activities designed to protect network. Specifically, these activities provides safe and secure network. Network security ensures reliable, usable and well integrated network usage. Effective network security targets a variety of threats and stops them from entering or spreading on network. Computer network security is simply a process or action implemented to detect as well as prevent unauthorized usage of your computer. It is a technique in the form of some kind of software; computer network security safeguards the networking infrastructure from illegitimate access, modification, malfunction, misuse, destruction, or unacceptable disclosure. It ensures protected environment and provides allowable

significant functions. Network Security is a very important part of corporate world today, even though it seems that vulnerabilities are not high, but serious damage can be caused from a remote point in a network.

Network security involves many activities like maintaining authorized access, that organizations, integrity and continuity of operations. An effective network security strategy requires identifying threats and then choosing the most effective set of tools to combat them.

Network security [4] is handled by a network administrator or system administrator. Every organization have their own security policy, for proper implementation [4] of security policy, software and hardware are needed. Network security protects specified resources in any organization. There are many different types of devices and mechanisms within the security environment to provide a layered approach of defense so that if an attacker [18] is able to sidestep one layer, nest layer stands in the way to protect the network.

Network security if applied in multiple layers; offers completely secure network. There is no single solution to security; there is need of multiple layers of security to protect system from a variety of threats. If one security layer fails, others will secure it. Hardware and software both are used for Network security. The software must be constantly updated and managed to protect from emerging threats. Network security involves various mechanisms and tools like firewalls, antivirus, intrusion detection system etc. usually a combination of tools gives infallible security solutions.

In order to strengthen the security, one cannot rely on any single tool. Hence, a firewall must be complemented by Intrusion Detection system.

Network security [2][8] refers to any activities designed to protect network. Specifically, these activities provides safe and secure network. Network security ensures reliable, usable and well integrated network usage. Effective network security targets a variety of threats and stops them from entering or spreading on network.

1.2.1. History of Network Security

Since the initiation of networked computers, security [4] has been the most important factor to consider. The birth of the internet takes place in 1969 when ARPANet (Advanced Research Projects Agency Network) is commissioned by the department of defense (DOD) for research in networking.

In the 1960s, the term “hacker” is coined by a couple of Massachusetts Institute of Technology (MIT) students. During the 1970s, the Telnet protocol was developed. This opened the door for public use of data networks that were originally restricted to government contractors and academic researchers.

In the late 80s government, universities and military connections increased with this network started to grow and security need was realized. In 1988 first worm come into sight on the ARPANET. A worm named “Morris Worm” was developed by a student. This worm could take advantage of the lack of intrusion prevention system and use vulnerabilities to copy itself. It spread by copying itself to connected computers and sending itself to a new location. The self-replicating Morris Worm did much to expose the vulnerabilities of networked computers - using so many resources that infected computers were rendered inoperable, and spreading quickly throughout the network. This made leaders in the network to take network threat seriously and subsequently development of countermeasures against network threats started.

Before the 90s, networks were relatively uncommon and the general public was not made-up of heavy internet users. During these times, security was not as critical - however, with more and more sensitive information being placed on networks, it would grow in importance. The network threat and risk was limited because network was small and network users were known to each other. When Internet became publics in nineties, the security apprehension increased enormously.

After 90s, there was tremendous growth in computer network. With this growth of network users in number and verity, security threats also increased. Public networks are being relied upon to deliver financial, personal and all type of information. Therefore, computer network security is highly essential.

Network security evolved a lot from its beginning. Due to, the evolution of computer networks and network applications there is a need to evolve computer network security continuously.

1.2.2. Network security components

Network security has many major components [18], which often includes:

- Anti-virus and anti-spyware,
- Firewall,
- Intrusion detection systems (IDS),

Figure 1.1 shows components of computer security.

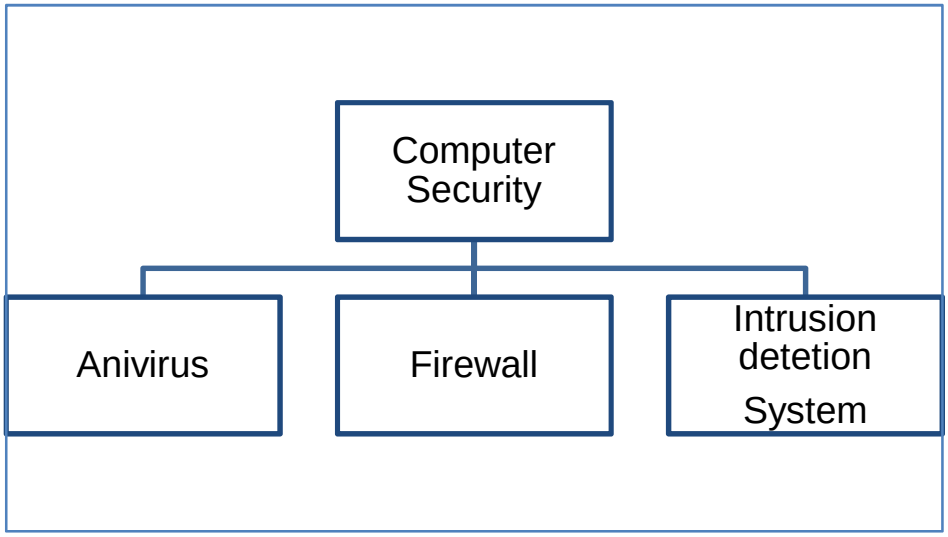


Figure 1.1 Network Security Components

- **Antivirus**

Anti-virus [3] [19] prevents and eliminate viruses. A virus programme saves computer from harmful software and damage. Antivirus software [21] protects the computer from infected files.

Antivirus detects the infections [19] in the system and repairs it, depending on the updated version. They will capture Infected of Files or email. Usual types of

infections are Trojan, Virus and Malware. Having anti-virus on computer is a necessity to evade malware or other virus attacks. Cyber attacks have become more complex and destructive with growth of network and information technology, therefore anti-virus now is not adequate. Their major function is to react only when the malware has already infiltrated the system.

- **Firewall**

A firewall [12] is a device installed between the internal and external network of an organization. It is intended to forward some packets and filter others. For example, a firewall may filter all incoming packets destined for a specific host or a specific server such as HTTP or it can be used to deny access to a specific host or a service in the organization.

A firewall is stalled away from the rest of the network so that no incoming requests get directly to the private network resource. Firewall gives security to protected computers if installed and configured properly. It filters network traffic based on following two methodologies:

- A firewall permits any traffic except what is specified as restricted. It relies on the type of firewall used, the source, the destination addresses, and the ports.
- A firewall refuses any traffic based on network layer; it refuses traffic which does not meet the specific criteria.

Advantages of firewall

- Firewalls can be configured as per organization's security policy.
- Firewalls can be configured to bar incoming traffic to POP and SNMP and to allow email access.
- Firewalls can secure from spam by blocking email services.
- Firewalls can be used to confine access to specific services.
- Firewall verifies the incoming and outgoing traffic against firewall rules.

- Firewall acts as a router in moving data between networks.
- Firewalls are excellent inspector. It inspects all traffic that passes through it.

Disadvantage of firewall

- A firewall is not able to detect sensitive information through social networking.
- Firewall only restricts defined traffic, but the traffic which is allowed through firewall if have any flaw, is not restricted.
- Firewalls efficiency completely depends on the rules which it is configured to enforce.
- Firewalls are not able stop attacks; if the traffic does not pass through them.
- Firewalls also can't secure Trojan attacks.
- Firewall fails against Tunnelling attempts.
- Firewalls are sometimes also not effective against network attacks. It cannot protect you from internal harm.

Security tool Firewalls act as a barrier between corporate (internal) networks and the outside world (Internet), and filter incoming traffic according to a security policy. Thus, a firewall provides a good amount of security lest sufficient protection due to the following facts:

- Access to the Internet occurs is not always through the firewall.
- It is not compulsory that threat originates only outside the firewall.
- Firewalls are subject to attack themselves.

Firewalls are not completely fail-safe. A firewall generally makes pass-deny decision on the basis of allowable network addresses. Intelligent firewalls may analyze the contents of packets of certain protocols but they may only identify the irregularity related to that protocol.

A common attack strategy is to utilize tunneling to bypass firewall protections. Tunneling is the practice of encapsulating a message in one protocol (that might be blocked by firewall filters) inside a second message. Thus, the inside message gets through as the firewall considers outer, encapsulating message harmless.

Consider an example of a bank security. Bank can be secured by allowing limited access control and by keeping fences in the world, but the biggest threat is the customers that are entering bank. So it is advisable to employ metal detectors to detect whether they are hididing any thing which can cause harm to security. In this example fences are like firewall, customers are network packet and metal detector are like intrusion detection system .

Firewalls are really good access control points, but they aren't really good for or designed to detect intrusions.

So there is strong need of another security component [5] along with antivirus and firewalls. Intrusion Detection Systems are the powerful systems when used along with antivirus and firewall gives a complete security mechanism.

Intrusion detection and prevention system

Intrusion prevention [7] sis a preemptive approach to network security used to identify potential threats and respond to them swiftly. IDS (Intrusion Detection System) [6] systems only detect an intrusion and alert to the administrator whereas IPS (intrusion prevention system) prevents system from intrusion attack. IPS slows down the network because of additional associated activities therefore usually avoided. IDS offers solution without affecting speed of network access.

According to Lappas [22] Intrusion detection in general, do not include prevention of intrusions. Like an intrusion detection system (IDS), an intrusion prevention system (IPS) monitors network traffic. Intrusion prevention process usually take more time than IDS . Hence, usually IDS is preferred over IPS.

1.2.3. Intrusion attack and its types

An intrusion attack [1] is realization of threat, the harmful action aiming to target and exploit the system vulnerability. Computer attacks may involve unauthorized access, destroying data; threaten the security computer or degrading its performance. Computer and network attacks have evolved greatly over the last few decades. The attacks are increasing in number and also improving in their strength and erudition.

- Attack motivation and objectives

Attack motivation can be understood by identifying what the attackers do. The main motivation of an attacker is to access to a system or data; the main motivation of the criminal is to get financial benefit. Other motivation factors are social, political gain. Mischievous human tendency is also motivate attack. The potential threat of cyber terrorism becoming inevitable due to the critical infrastructures that is potentially vulnerable [15] [16] . It is easy to attack due to growth of network.

A) Types of intrusion attack

Intrusion attack [14] [18] can be categorized into four major types DoS, Probe, U2R, R2L.figure 1.2 shows types of attacks.

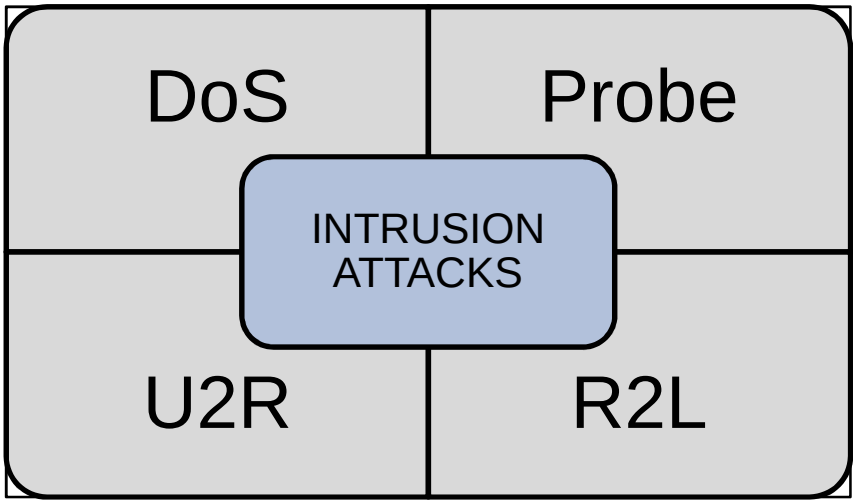


Figure 1.2 Types of intrusion attack

- **DoS attack**

In a denial of service [15] attack, an attacker makes a resource on a network either unavailable to justifiable users. DoS attacks make system processes very busy and occupied with unwanted, unidentified processes. It attacks on the resource like network bandwidth, computer memory or computing power. There are many different types of DoS attacks. For example attack can deny access to a machine on, a network. The DoS attacks [25] [26] are meant to force the target to stop the service(s) that is (are) provided by flooding it with probes illegitimate requests.

- **Probe attack**

Probe attacks [16] are often the first step of all other attacks. Probe attacks are used to collect information about the targeted computer network or a definite machine on computer network. Network probes are most important for attacker because through this only they find vulnerabilities present on his target machine or network. That is the reason why it is critical to detect this type of attacks. Mostly all administrator uses probe to check machines on a network, so it is difficult to detect which one is legitimate user and which one is attacker. So it is also difficult to distinguish attacks from regular actions.

The probe attacks are meant to obtain information about the target network from a source that is usually external to the targeted network. Probing is an attack in which the hacker scans a machine or a networking device in order to determine weaknesses or vulnerabilities that may later be exploited so as to compromise the system.

- **U2R**

The U2R [16] attacks are difficult to arrest because it involve the semantic details that are very difficult to capture at an early stage. Initially attacker starts off on the system with a normal user account and then tries to get super user privileges rules by abusing vulnerabilities.

In a User to Root attack, an attacker starts a session on a computer as a normal user with restricted rights and by exploiting some vulnerability on the software installed on the system, the user can raise his privilege. The purpose of this class of attack is obviously to obtain administrator rights on the attacked computer in order to have full control of it. There are several different types of U2R attacks. Buffer overflow is undoubtedly the major vulnerability used by hackers when trying to obtain privileged rights on a computer.

- **R2L**

Most challenging attacks are R2L attacks [16] they are very difficult to detect because they involve the network level and the host level features. A remote to user attack is an attack in which a user sends packets to a machine over the internet, which attacker does not have access to in order to expose the machines vulnerabilities and exploit privileges which a local user would have on the computer **.**

In a Remote to Local attack, the attacker starts from a session on a computer outside of the targeted network and exploits vulnerability in order to gain access to a computer on the local network. A precondition that must be fulfilled is the ability for the attacker to send network packets to the victim host. Usually, but not always, Remote to Local attacks are combined with U2R attacks permitting the attacker to get full access of a remote machine which is part of a other network than the network of the attacker.

B) Details of some common attacks

- Back:

This attack is initiated against an apache Web server, which is flooded with requests containing a large number of front-slash (/) characters in the URL description. As the server tries to process all these requests, it becomes unable to process other genuine requests and hence, it denies service to its customers.

- Smurf Attack:

In a ‘smurf’ attack is a type of DoS attack, in this attack many ICMP echo-reply packets are bombarded on attacked machine. This attack throw many ICMP echo-request packets to the broadcast address of many subnets Every machine that belongs to any of these subnets responds by sending ICMP ‘echo-reply’ packets to the victim. These packets contain the victim's address as the source IP address. Smurf attacks are very hazardous, because they are strongly distributed attacks.

- Teardrop:

Many times a packet is broken into smaller fragments while travelling from the source machine to the destination machine. A Teardrop attack creates a stream of IP fragments with their offset field overloaded. The destination host that tries to reassemble these malformed fragments eventually crashes or reboots.

- Land:

The Land a very common DoS (Denial of Service) attack works by sending a spoofed packet with the SYN flag - used in a ‘handshake’ between a client and a host - set from a host to any port that is open and listening. If the packet is programmed to have the same destination and source IP address, when it is sent to a machine, via IP spoofing, the transmission can fool the machine into thinking it is sending itself a message, which, depending on the operating system, will crash the machine.

- Neptune (SYN Flood):

Neptune (SYN Flood) is a attack to which every TCP/IP implementation is vulnerable. Each half-open TCP connection made to a machine causes the 'tcpd' server to add a record to the data structure that stores information describing all pending connections. The data structure which is used for this work is of finite size,

and it can be made to overflow by intentionally creating too many partially-open connections. The half-open connections data structure on the victim server system will eventually fill and the system will be unable to accept any new incoming connections until the table is emptied out.

- Ping of Death(POD):

In Ping of Death attacks is the DoS attack in which attacker creates a packet of size more than IP protocol limit (more than 65,536 bytes). This packet can cause different kinds of damage like rebooting and crashing of the machine that receives it.

- Portsweep

A port sweep attack scans multiple hosts for one port. For example port 80 is usually scanned for all the addresses in a 24 bit address space. To portsweep is for one listening port scanning multiple host. It searches for a specific service, like SQL-based computer worm may portsweep looking for hosts listening on TCP port.

- NMAP

Nmap is the a type of port scanner. Nmap has a large list of parameters and perform following :

  - Host discovery – Identifying hosts on a network. For example, listing the hosts that respond to pings or have a particular port open.
  - Port scanning – Enumerating the open ports on target hosts.
  - Version detection – Interrogating network services on remote devices to determine application name and version number.
  - OS detection – Determining the operating system and hardware characteristics of network devices.
  - Scriptable interaction with the target – using Nmap Scripting Engine (NSE) and Lua programmeming language.

- Nmap can provide further information on targets, including reverse DNS names, device types, and MAC addresses.

- SATAN

SATAN (Security Administrator Tool for Analyzing Networks) remotely probes systems through the network. Satan stores its findings in a database. SATAN is a publicly available tool that probes a network for security vulnerabilities and mis-configurations. It is created to be used by administrators but often used by attackers to search for vulnerabilities on a network. Information provided by SATAN could be useful to an attacker in performing an attack.

Internet community uses a shareware version of SATAN extensively. SATAN collects data from the named hosts, that it discovers while probing a primary host. A primary target can be a host name, a host address, or a network number. SATAN can generate reports of hosts by type, service, vulnerability and by trust relationship. it also gives details of vulnerabilities and way to handle and remove them.

- phf Attack

A script named ‘phf’ can be. The legitimate use of the phf script is to update the people directory, which is installed by default in the cgi-bin directory. It is used to perform an attack on the web server many times .The script’s behaviour changes if used with the ‘0a’ character in the URL when calling the script. To perform an attack, the attacker appends ‘0a’ to the URL along with some other UNIX command.

- Buffer overflows

There were four buffer overflow attacks on eject, fdformat, ffbconfig , and ps programmes. The attacks on the first three programmes exploited a buffer overflow condition to execute a shell with root privileges. The specification used to monitor setuid to root programmes could easily detect these attacks by detecting oversized arguments and the execution of a shell. The ps attack was significantly more complex than the other three buffer overflow attacks. For one thing, it used a buffer overflow

in the static area, rather than the more common stack buffer overflow. Thus, it is difficult to detect. Second, instead of shell program it used a chmod system call to effect damage. chmod operation is itself unusual, and it is not permitted by generic specification (except on certain files).

- Ftp-write attack

The ftp-write attack is a R2L (remote to local) user attack that takes advantage of a common anonymous ftp misconfiguration. The ftp directory and its subdirectories should not be owned by the ftp account or be in the same group as the ftp account. If any of these directories are owned by ftp or are in the same group as the ftp account and are not write protected, an intruder will be able to add files and eventually gain local access to the system. This attack is easy to attack due to the site-specific policy that no file could be written in ftp directory.

- Warez attacks

There are two types of warez attacks ; warezmaster and warezclient . warezmaster attack logs into an unidentified FTP site and creates a file or a hidden directory. In warezclient attack, the file previously down loaded by the warezmaster is uploaded. This attack could be easily captured by the specifications which encoded the site-specific policy of disallowing any writes to the FTP directory.

1.2.4. Intrusion detection system

Intrusion detection [17] is viable and practical approach for providing a different notion to security of computer and network systems

Intrusion detection systems (IDSs) are monitoring system that have been added to the wall of security in order to prevent malicious activity on a system .Intrusion Detection is the inexorable active efforts in discovering or detecting the presence of intrusion attack. In the field of computer network security, significance of Intrusion detection system (IDS) is well established. Intrusion detection is a new, recent approach for providing a sense of security in existing computers and data networks.

Intrusion Detection Systems (IDS) [11] [13] are the second layer of defense. It detects the presence of attacks within traffic that flows in through the holes punched into the firewall. Intrusion detection is the process of which supervises the events occurring in a computer system or network to analyze them for signs of intrusion.

To understand the difference between firewalls and IDS, firewall only restrict defined traffic whereas IDS monitors that traffic which flow through firewall. So IDS is next layer of security.

- **Why we need IDS ?**

To answer this question, we need to understand why intruders can get into the system. There are various reasons of which the prominent ones are:

- Software bugs – they can be buffer overflows, unexpected combinations, unhandled inputs, race conditions etc. Software has bugs because programmers cannot track down and eliminate all possible holes.

- Password Cracking – hackers have over the time developed numerous ways to break into systems by knowing passwords that were really weak, or by making dictionary & brute force attacks.

- Design flaws – many systems that were developed early were never designed to handle the wide scale intrusion that is there today. These include TCP/IP protocol flaws, operating system flaws etc.

- Sniffing unsecured traffic – traffic on the Internet is not encrypted. Hackers can use programmers that can get sensitive information from packets over the network. These include the packet sniffers, port scanners etc.

A firewall cannot always handle attacks directed to exploit these flaws. Hence, we require IDS which can logically complement the firewall.

- **Paradigms in intrusion detection**

Intrusion detection system can be categorized in misuse detection and anomaly detection [24].

A) Misuse detection model

Signatures [20] are patterns corresponding to known attacks or misuses of systems. They may be simple (character string matching looking for a single term or command) or complex (security state transition written as a formal mathematical expression). In general, a signature can be concerned with a process (the execution of a particular command) or an outcome (the acquisition of a root shell.) Signature analysis is pattern matching of system settings and user activities against a database of known attacks. The database of known attacks (pattern file of attack signatures) is analogous to the virus definitions file of a virus scanner.

Most commercial intrusion detection products perform signature analysis against a vendor-supplied database of known attacks. Additional signatures specified by the customer can also be added as part of the intrusion detection system configuration process. These databases are periodically updated. One advantage of signature analysis is that it allows sensors to collect a more tightly targeted set of system data, thereby reducing system overhead.

The strength of signature analysis depends upon the quality, comprehensiveness, and timeliness of the attack signature in the IDS's search engine. Poorly defined signature can cause false positives [23] means normal packet is identified as attack or attack is shown normal packet. Pattern matching tools are excellent at detecting known attacks, but perform poorly when confronted with a fresh assault, or a modified old one.

B) Anomaly detection model

Statistical analysis [20] finds deviations from normal patterns of behaviour. Statistical profiles are created for system objects (e.g., users, files, directories, devices, etc.) by measuring various attributes of normal use (e.g., number of accesses, number of times an operation fails, time of day, etc.). Mean frequencies and measures of variability are calculated for each type of normal usage. Possible intrusions are signalled when

observed values fall outside the normal range. For example, statistical analysis might signal an unusual event if an accountant who had never previously logged into the network outside the hours of 8 AM to 6 PM was to access the system at 2 AM.

Anomaly Detection in IDS includes:

- Threshold detection detecting abnormal activity on the server or network, for example abnormal consumption of the CPU for one server, or abnormal saturation of the network .
- Statistical measures, learned from historical values
- Rule-based measures, with expert systems
- Neural Networks or Genetic algorithms

In principle, an Anomaly Detection IDS 'learns' what constitutes 'normal' network traffic, developing sets of models that are updated over time. These models are then applied against new traffic, and traffic that doesn't match the model of 'normal' is flagged as suspicious. Anomaly Detection IDS **[25][27]** is very promising, but they require proper training. If not trained properly it may give false alarms.

Advantages:

The system may detect unknown attacks also with high accuracy. Statistical methods may allow one to detect more complex attacks, such as those that occur over extended periods.

Disadvantages:

Anomaly detection system if not trained properly; can accept an attack activity as normal by gradually changing behavior over time. The possibility of false alarms is much greater in such type of detectors. Statistical detectors do not deal well with changes in user activities.

It is also difficult of define rules. Each protocol being analyzed must be defined, implemented and tested for accuracy. The rule development process is also complex.

Moreover, detailed knowledge of normal network behaviour must be constructed and transferred into the engine memory for detection to occur correctly. On the other hand, once a protocol has been built and a behaviour defined, the engine can scale more quickly and easily than the signature-based model because a new signature does not have to be created for every attack and potential variant.

Hence, IDS is broadly categorized into misuse detection and anomaly detection. Generally anomaly based IDS perform better than misuse detection.

1.3. Need and significance of study

In this era of IT (Information Technology), IT industries are growing extensively. Tremendous growth network facilities and services make work easy but raise issue of computer network security. One of the greatest threats to computer network security is intrusion based security attack.

Intrusion based security attacks causes serious harm to computer network, therefore significance of intrusion detection system is widely accepted. Intrusion detection field is evolving continuously as attack methods and influence is increasing continuously. Therefore there is a need to empower intrusion detection systems to strengthen computer network security. In general, as the organizational network grows to accommodate changing needs, more robust technology solutions are required.

Therefore researcher is identifying the domains of computer network security and their implementation by the IT industrial units order to find out effectual methods for detection of security threats.

To address computer network security needs, data mining research is providing better results. But there is need to explore this field to get better technical solutions. This research explores applicability and usability of data mining techniques to computer network security. Various data mining techniques need to be investigated for this. This research is intended to get security solution by performing experiments by data mining techniques.

1.4. Organization of the thesis

This thesis is structured into seven chapters.

Chapter 1:

In the first chapter researcher has given brief introduction to this research work. This further gives need and significance of study.

Chapter 2:

The second chapter discusses about review of literature. Researcher reviewed different related literatures in order to have detailed understanding on the present research. Along with study of literature popular IDS products and their features are compared based on the available literature and documentation.

Chapter 3:

The third chapter discusses Research Methodology.

Chapter 4:

The forth chapter analyses and interprets the data collected through survey. This chapter also deals with testing of hypothesis.

Chapter 5:

This chapter provides a comprehensive discussion about the experimentation part of this thesis. This chapter discusses the results of experiments and finally provides a data mining framework for intrusion detection. Developed framework SIDDM (Systematic Intrusion Detection using Data Mining) is elaborated.

Chapter 6:

This chapter discusses observations and findings of the study

Chapter 7:

This chapter gives conclusion to this research work. Features and advantages of proposed model are presented. This chapter gives overview, to complete research works , gives suggestion and finally discuss scope for future work.

Appendix

This contents the Questionnaire for survey in Annexure 1. Result of data mining experimentation Decision tree rule are given in Annexure 2. List Publication based on research by researcher is given in Annexure 3.

Bibliography

This contains the references of material referred by the researcher to study network security, data mining theoretical aspects.

1.5. Chapter References

1. Adeyinka, O.,(2008), "Internet Attack Methods and Internet Security Technology Modeling & Simulation" , AICMS 08. Second Asia International Conference on,vol., no., pp.77-82.

2. Ajibuwa F. O. ,(2006), " Data and Information Security in Modern Day Businesses" ,Published M.Sc. dissertation, Atlantic International University, U.S. ,from http://www.aiu.edu/publications/student/ english/Data and Information Security in Modern Day Businesses thesis.html .

3. An Introduction to Computer Security: The NIST Handbook ,Special Publication 800-12.

4. Antivirus - how-antivirus-software-works , http://www.howtogeek.com /125650/ htg-explains-how-antivirus-software-works.

5. Basta A., Halton W.,(2003) “Computer Security- Concepts, Issues and Implementation”, New Delhi: Course technology/cengage Learning.

6. Bhavya Daya ,(2010), “Network Security: History, Importance, and Future”, University of Florida Department of Electrical and Computer Engineering.

7. Bishwanath mukharjee L.Todd heberlien,(1994), “Network Intrusion Detection” ,IEEE.

8. Carl F.,(2003), “Intrusion Detection and Prevention”, McGraw-Hill, Osborne Media.

9. Counteract edge for threat prevention www.forescout.com/product /counteract-edge, Accessed date Jan 2012

10. Curtin M., (1997),“Introduction to Network Security,” http://www.interhack.net/ pubs/network-security.

11. Everything you need to know about network security, www.axent.com , last accessed 2012.

12. Heberlein L., Dias G., Levitt K., Mukherjiee B., Wood J., and Wolber D.,(1990), “A Network Security Monitor” , Proc., IEEE Symposium on Research in Security and Privacy, Oakland, CA, pp.196-304.

13. John McHugh, Alan Christie, and Julia Allen ,(2000), “The Role of Intrusion Detection Systems”, IEEE SOFTWARE .

14. John Wack, Ken Cutler, Jamie Pole,(2002), “Guidelines on Firewalls and Firewall Policy ” ,Recommendations of the National Institute of Standards and Technology.

15. Karen S. and Peter M.,(2007), “Guide to Intrusion Detection and Prevention Systems”, National Institute of Standards and Technology, Department of Commerce, USA.

16. Kendall, K.,(1999) " A database of computer attacks for the evaluation of intrusion detection systems" , Masters thesis, Massachusetts Institute of Technology.

17. Kevin J. Houle,(2001), "Trends in Denial of Service Attack Technology", CERT Coordination Center.

18. Marin, G.A.,(2005), "Network security basics Security & Privacy" ,, IEEE , vol.3, no.6, pp. 68-72.

19. Mithcell Rowton,(2005), "Introduction to Network Security Intrusion Detection" , December 2005.

20. Roman V. Yampolskiy and Venu Govindaraju, (2007),"Computer Security: a Survey of Methods and Systems ", Journal of Computer Science 3 (7): 478-486.

21. Saumil Shah, "the Anti Virus Book", The Tata McGraw-Hill Publishing Company Ltd. http://saumil.net/antivirus

22. T. Lappas and K. P. ,(2007), "Data Mining Techniques for (Network) Intrusion Detection System" , 2007.

23. Tadeusz Pietraszek, Axel Tanner, (2005),"Data mining and machine learning—Towards reducing false positives in intrusion detection" ,2005.

24. Wenke Lee,(2002), "Applying Data Mining to Intrusion Detection: the Quest for Automation, Efficiency, and Credibility", SIGKDD.

25. Whitman M. E. & Mattord H. J. ,(2007), "Principles of Information Security" (2nd ed.), New Delhi: Thomson Learning/Course Technology.

26. Wu Junqi1, Hu Zhengbing.,(2008), " Study of Intrusion Detection Systems (IDSs) in Network Security".,ISSN 978-1-4244-2108-4/08/, 2008 IEEE

27. Zachary Miller, William Deitrick, Wei Hu, (2011), "Anomalous Network Packet Detection Using Data Stream Mining" , scientific research, Journal of Information Security.

Chapter 2

Review of Literature

2.1. Introduction

One of most essential step, in any research is to take a review of available literature pertaining to the research subject. A review of literature facilitates the researcher to determine the specific subject area. A review of literature also gives in-depth knowledge related to the subject matter, helps to reveal the gaps remained in the available literature, and provides direction and guidance. It sometimes gives different perspectives to look at the particular question. It helps to understand the importance, background and the present situation related to the subject selected for the research. It provides background of the earlier studies in the similar subject. It also gives a confirmation that the present study has already taken note of what others have done and written in selected area. Therefore, it is necessary to review all kinds of literature related to the subject matter. A review of the work in the intrusion detection domain related to this research's approach is presented.

2.2. Importance of Network Security

In this section some of the prominent work related to network security is reviewed.

1. **Bishwanath Mukharjee** have published paper entitled **"Network Intrusion Detection" [2]**

   This research paper proposes role of intrusion detection system for secure computer network. This is one of milestone paper in network security. This elaborates intrusion attack and their attributes. Further this paper compares various intrusion detection systems products on system organization and

capability to detect intrusion. IDS products like Computer watch, discovery, haystack , IDES ,ISOA ,MIDAS ,wisom and sense, NADIR Network anomaly ,NSM Network security monitor, DIDS distributed intrusion detection system are reviewed . this paper is important because it gives idea about anomaly detection based IDS. An algorithm to detect intrusion is provided, which is based on weighted intrusion score. Prototype suggested can be used for development of IDS software.

2. **Dorothy E. Denning** have published paper entitled **"An Intrusion-Detection Model" [12]**

   The IDS model proposed in this research work is based on the hypothesis that security violations can be detected by monitoring a system's audit records for abnormal patterns of system usage. This model analyses subject's behavior with respect to object behavior. Knowledge is acquired using rules about this behavior .audit records are used to detect and analyze anomalous behavior. The model allows intrusions to be detected without knowing about the flaws in the target system that allowed the intrusion to take place, and without necessarily observing the particular action that exploits the flaw.

3. **Tim lane** have published thesis entitled **"information security management in Australian universities: an exploratory analysis" [57]**

   This thesis analyzes security issues in Australian universities. This describes challenges and methods of information security management are discussed. it further elaborates achievable improvements in information security management. This is a survey based research.

4. **Adeyinka, O.** published paper entitled **"Internet Attack Methods and Internet Security Technology Modeling & Simulation".[1]**

   In this paper four major security attributes are identified, these attributes are Confidentiality, Integrity, privacy and Availability. This paper discusses attribute wise attack methods and tools to maintain security. Attack methods identified on

Confidentiality are Eavesdropping, Hacking, Phishing, DoS and IP Spoofing. To maintain confidentiality tools like IDS, Firewall, Cryptographic, Systems, IPSec and SSL are suggested. Integrity loss can cause by Viruses, Worms, Trojans, Eavesdropping, DoS and IP Spoofing. to provide integrity IDS, Firewall ,Anti-Malware Software, IPSecand SSL. Privacy can loss by Email bombing, Spamming, Hacking, DoS and Cookies. To maintain privacy IDS, Firewall, Anti-Malware Software, IPSecand SSL can be used. DoS, Email bombing, Spamming and Systems Boot Record Infectors causes loss of availability. To manage availability IDS, Anti-Malware Software and Firewall are suggested.

5. **Roman V. Yampolskiy et al.** have published paper on **"Computer Security: a Survey of Methods and Systems" [48]**

   In this research attack, bugs and viruses are analyzed. Attackers are classified on basis of type of access used to attack system. Various security methods and issues are discussed. This paper suggests that different solutions collectively give effective solution against different types of attacks. Security must be continuously monitored using efficient tools.

6. **Bhavya Daya** have published paper entitled **"Network Security: History, Importance, and Future" [3]**

   The paper explains network security history in detail. This paper discusses popular internet architecture and related security issues. This paper adequately summarizes history of internet and computer network security. Paper shows development in network security is categorized as per software development and hardware development. According to researcher for effective security management firewalls, intrusion detection and authentication mechanisms must be combined with use of IPv6. Paper discusses how modified internet Architecture and various tools can make network security efficient.

7. **Chia-Mei Chen** in this paper entitled **"An efficient network intrusion detection" [6]**

Chia Mei Chen proposed a Lightweight Network Intrusion Detection system for detecting such attacks on Telnet traffic. It characterizes normal traffic behavior and computes the anomaly score of a packet based on the deviation from the normal behavior. Instead of processing all traffic packets, an efficient filtering scheme proposed in the study can reduce system workload and only 0.3% of the original traffic volume is examined for anomaly. According to the performance comparisons with other network-based IDS, model proposed is the efficient on detection rate and workload reduction.

8. **Hulus onder** have published paper entitled **"A security management system design" [26]**

   This paper presents the difficulties of managing the security for an enterprise network. This thesis explains in detail about better management of security and issues related to higher management. This further proposes a Security Management System for network security management. Security management system suggested is easy to use, flexible and scalable. This security management work has some drawbacks like it is not rule based and do not provide artificial intelligence techniques.

9. **S.S. Joshi** have published doctorial dissertation **"A Study of Information Security Policies in Selected IT Companies in Pune City" [52]**

   This thesis presents survey and analysis of Information Security Policies in IT Companies of Pune City. According to this study Information security policy is very essential for every IT organization and most of IT companies in Pune effectively employed information security policies. These policies are implemented regularly. These policies can be categorized into administrative policies and Technical policies. Employees of Pune companies know administrative policies are more than the technical one. Technical policies are restricted to specific domain therefore not implemented by all types of the employees. Effective management of Information security is critical for the success and survival of any type of IT organization. Paper identifies the status of information security policies in Pune city IT companies.

2.3. Data mining methods for intrusion detection

A literature review of existing techniques relating to work in this thesis is presented. In particular, this review looked at some of the work done in the area of application of data mining technology for intrusion detection.

1. **Daniel Barbara et al.** have published paper entitled **"ADAM: Detecting Intrusions by Data Mining" [9]**

   This paper presents a model ADAM (Audit Data Analysis and Mining) this data mining model was proposed in 2001. ADAM uses a combination of association rules mining and classification to discover attacks in a TCP dump audit trail. ADAM is a two stage system; the first stage is a rule mining stage that creates a network traffic profile in the form of association rules based on attack free training data. Another component of this stage, fed with training data including attacks along with the normal profile rules, generates attack rules dynamically. A third component of this stage extracts other features from the training data. The outputs of the last two components (i.e., attack rules and extracted features) are used as a training set for the second stage - a classifier based on pseudo-Bayes estimators. The purpose of the classifier is to further analyze the attacks predictions before passing it onto the security expert.

2. **Wenke Lee** have published thesis entitled **"A Data Mining Framework for Building Intrusion Detection Models " [59]**

   Lee presented a model JAM. The main idea in JAM is to generate classifiers using a rule learning programme on training data sets of system usage. The output from the classifier, a set of classification rules, is used to recognize anomalies and detect known intrusions. The main difference between JAM and ADAM is that JAM uses misuse detection system by learning the characterization of the attacks whereas ADAM uses an anomaly-based approach. Finally, the authors mention the use of a Meta-detection model that describes how multiple base classifiers [35] can be combined in order to exploit combined evidence of multiple traffic patterns.

3. **Levent Ertoz et al.** have published paper entitled **"The MINDS – Minnesota Intrusion Detection System"** [37]

   This paper presents a model called MINDS -Minnesota Intrusion Detection System. It is a data mining based system for detecting network intrusions. It uses a suite of data mining techniques to automatically detect attacks against computer networks and systems. Density based outlier detection scheme used in its anomaly detection module. Specific contributions of MINDS are: (i) an unsupervised anomaly detection technique that assigns a score to each network connection that reflects how anomalous the connection is, and (ii) an association pattern analysis based module that summarizes those network connections that are ranked highly anomalous by the anomaly detection module. It is a network level anomaly detection system that also incorporates a signature component. Intrusion detection is near real time and not instantaneous. Further, the number of alarms generated from each 10 minutes data is in thousands. In addition, the new signature creation is still a manual process.

4. **Fangfei Weng et al.** have published paper entitled **"An Intrusion Detection System Based on the Clustering Ensemble"** [17]

   This paper presents an unsupervised anomaly detection system based on the clustering ensemble. The system is based on the multiple runs of K-means to accumulate evidence to avoid the false classification of anomalistic data; then using single-link t o construct the hierarchical clustering tree to get the ultimate clustering result. Paper introduces a new clustering algorithm, the Evidence Accumulation (EA) for intrusion detection based on the concept of Clustering Ensemble, constructing an Intrusion Detection System based on Evidence Accumulation (EAIDS).

5. **Herkshop S. et al.** have published paper entitled **"A data mining approach to host based intrusion detection"** [25]

   This paper presents several problems inherent in developing and deploying a real-time data mining-based IDS. Several approaches are discussed like unsupervised

anomaly detection algorithms, ensembles of classification models is discussed. Further this paper shows implementation of feature extraction and construction algorithms for labeled audit data. The architecture consists of sensors, detectors, a data warehouse, and a model generation component. The computational costs of features are analyzed and a multiple-model cost based approach is used to produce detection models with low cost and high accuracy. Paper also presents a distributed architecture for evaluating cost-sensitive models in real time. By using adaptive learning algorithms ,usability is improved . This work suggests anomaly detection work using unsupervised model. As suggested model is unsupervised therefore model is less dependent on labeled data.

6. **T. Lappas et al.** have published paper entitled **"Data Mining Techniques for (Network) Intrusion Detection System"** **[55]**

   T. Lappas has presented a survey of the various data mining techniques that have been proposed by many models towards the enhancement of IDSs. Machine learning techniques like inductive rule learning, support vector machine, genetic algorithm, neural network and clustering methods are discussed in detail. In this paper, statistical techniques which are applicable to intrusion detection are also discussed. Author presented a new technique bi-clustering for intrusion detection. Intrusion detection taxonomy is presented on intrusion detection approaches, data sources, structure, protected system, analysis timing and attack behavior.

7. **Kamran Shafi** have published paper entitled **"An Online and Adaptive Signature-based Approach for Intrusion Detection Using Learning Classifier Systems"** **[32]**

   This thesis proposes Distance Based Technique to improve UCS (supervised classifier system) performance. This thesis introduces Subsumption operators to resolve overlapping and redundancies among the signatures. The methodology suggested is based on supervised learning algorithm. The rule learning systems developed in this thesis uses domain knowledge and do not provide feature selection procedures. This suggests signature extraction method of adaptively learning maximally rules.

8. **Flora S. Tsai** have published paper entitled **"Network Intrusion Detection Using Association Rules" [18]**

   This paper presents to detect Intrusion using association rules this system generates attack rules that will detect the attacks in network audit data using anomaly detection. This shows how the modified association rules algorithm is capable of detecting network intrusions. The system can show the overall results that display the item set versus the attack category accuracy that allows the user or administrator to filter out those unnecessary item sets and concentrate on those item sets that produce more accurate results.

9. **Dewan Md. Farid et al.** have published paper entitled **"Attacks Classification in Adaptive Intrusion Detection using Decision Tree" [11]**

   This paper presents, a new learning algorithm for anomaly based network intrusion detection using decision tree, which adjusts the weights of dataset based on probabilities and split the dataset into sub-dataset until all the sub-dataset belongs to the same class. In this approach weights of every example change based on posterior probability are considered.

10. **Ghanshyam Prasad Dube et al.** have published paper entitled **" A Novel Approach to Intrusion Detection System using Rough Set Theory and Incremental SVM" [22]**

    This paper proposes the use of RST (Rough Set Theory) and Incremental SVM (Support Vector Machine) for detection of intrusions. First, RST is used to preprocess the data and reduce the dimensions. Next, the features were selected by RST will be sent to SVM model to learn and test respectively. The method is effective to decrease the space density of data. This method, overcomes the shortages of SVM time-consuming of training and massive dataset storage.

11. **Huu Hoa Nguyen et al.** have published paper entitled **"An Efficient Fuzzy Clustering-Based Approach for Intrusion Detection" [27]**

This paper presents the idea to take useful information exploited from fuzzy clustering into account for the process of building IDS. The incorporation of cluster features resulting from a fuzzy clustering into the training process is used in this paper. Experimental results based on various data mining methods like C4.5 decision tree, Boosting, Bagging, SVM inducer with Polynomial Kernel, SVM inducer with Radial Basic Function Kernel are compared with CFC algorithm. This demonstrates efficiency of suggested methods.

12. **Gunja Ambica et al.** , have published paper entitled **" Robust Data Clustering Algorithms for Network Intrusion Detection" [24]**

    This paper presents an approach to detect intrusion based on unsupervised data mining frame work. In this framework, intrusion detection is achieved using clustering techniques. a method to lessen the noise in the data set using improved K-means is presented . This system use K-means, FCM and Improved K-means data mining algorithms are used to progress the performance of intrusion detection. Network traffic is usually large and have possibility of various types of attack so methods for detection must be accurate. By the more accurate method of finding k clustering center, and anomaly detection model was presented to get better detection effect.

13. **Nagaraju Devarakonda et al**. have published paper **"Intrusion Detection System using Bayesian Network and Hidden Markov Model" [43]**

    The paper presents IDS model based on Bayesian Network and the Hidden Markov Model (HMM) method with KDDCUP dataset. The IDS framework has been designed with various levels of processing such as model learning with training data and constructing the Bayesian Network and this structure has been used as HMM state transition diagram. The preprocessed KDDCUP dataset has been used to train and test the model. The IDS model has been trained and tested for normal and attack type connection records separately. The results evince that the performance of the model is of high order for classification of normal and intrusions attacks.

14. **Shyara Taruna R**. **et al.** have published paper entitled **"Enhanced Naïve Bayes Algorithm for Intrusion Detection in Data Mining"** [49]

    This paper proposes a new method of Naïve Bayes Algorithm. This presents how effective detection rate can be obtained through supervised approach for anomaly detection. The performance of our proposed algorithm is tested by employing KDD99 benchmark network intrusion detection dataset. The experimental results proved that it reduces false positives for different types of network. Suitability of naïve bayes for analyzing large numbers of network logs or audit data is demonstrated.

15. **G.V. Nadiammai et al.** have published paper entitled **"Effective approach toward Intrusion Detection System using data mining techniques"** [21]

    This paper proposes data mining method for network intrusion detection data mining technique hybrid PSO is used. This paper is based on semi-supervised model training concept. The labeled training data are applied to the SVM classifiers are used for model is generation; this model is able to detect Anomaly in network packet along with snort.

2.4. Data mining Theoretical background

Data mining [29] is the process of automatically scanning huge amount of data and searching available patterns in it. Storing large amount of data is useful only when we extract useful information from it. Data mining deals with large volume of data to extract meaningful information. Data mining refers to extracting or mining knowledge from large amounts of data [34] . In data mining, algorithms seek out patterns and rules within the data from which sets of rules are derived. Algorithms can automatically classify the data based on similarities (rules and patterns) obtained between the training and the testing data set.

Data mining [9] is the process of discovering patterns in data, either automatically or semi-automatically. The patterns discovered must be meaningful in that they lead to some advantage, usually financial advantages. Data mining combines concepts, algorithms and tools. It has derived concept from machine learning and statistics for the analysis of very large datasets. Data mining gain insights, understanding of data and provides actionable knowledge. Data mining provides capability to predict the outcome of a future observation. Other than predicting future observation, data mining is also useful for summarizing the underlying relationship in data.

Data mining can mine data from different data storage like text data, databases, data warehouse, transactional data, multimedia data, stream, spatiotemporal, time-series, sequence, and web, multi-media, graphs & social and information networks etc. The field of data mining grew out of the limitations of current data analysis techniques in handling challenges posed by these new types of datasets.

Today, data mining has grown so vast that they can be used in many areas like financial analysis, customer management, risk management, predicting costs of corporate expense claims, healthcare, insurance, process control in manufacturing and in other fields. This thesis illustrates how data mining is also applicable in computer security management.

Data mining analyzes data from different perspective and summarizes it into useful information. It also analyzes data from many different dimensions, then it categorizes and summarizes the relationships identified. Technically, data mining is the process of finding correlations or patterns among various fields in large datasets. The current developments in data mining contributed a wide variety of algorithms, drawn from the fields of statistics, pattern recognition, machine learning, and database which is useful for technology adaptation and usage.

Data mining is able to predict important things in advance. That technique that is used to perform these feats is called modeling. Modeling is simply the act of building a model. A model is a set of rules, examples or mathematical relationships. Model is built on data from situations where the outcome is known and then this model is applied to other situations where the outcome is not known. Modeling techniques

have been around for centuries, but techniques of huge data storage, data communication capabilities and ability to process complex data is recently developed, so modeling is applicable to new areas.

As a simple example of building a data mining model [9], consider the director of educational institute. He/she would like to focus results and educational quality of his institute. Large amount of student data is usually available at all the institutes. He knows a lot about his students, but it is impossible to discern the common characteristics of his students. From the existing database of students, which contains information such as age, sex, academic history, continuous assessment details, family background etc., he can use data mining tools for discovering useful patterns such as relation between student's previous academic performance with entrance examination score , continuous assessment data with their final examination results, or predicting about failure cases, the placement package received by a student, establishing association between two elective subjects registered by a student in a semester, number of international students admitting to the institute. Data mining will be very helpful for such analysis of the large amount of data, which in turn will help for academic performance improvements, planning, promotional activities etc.

Data mining [9] is primarily used today by companies to acquire information about their customers .data mining also enables these companies to determine relationships among "internal" factors such as price, product positioning, or staff skills, and "external" factors such as economic indicators, competition, and customer demographics.

2.4.1. Data mining and Knowledge discovery

Data Mining is a step in KDD [41] process which uses specific algorithms for extracting patterns (models) from data. The term KDD refers to the overall process of discovering useful knowledge from data. The KDD process has other steps like data preparation, data selection, data cleaning etc. At first, data is obtained from various data sources, then data preprocessing like data cleaning and data integration is

applied. This creates data warehouse. From data warehouse task relevant data is taken and data mining is applied on this. Data mining applies pattern evaluation to extract knowledge. Therefore, Data mining plays an essential role in the knowledge discovery process.

The KDD process refers to the whole process of changing low level data into high level knowledge which is automated or semi-automated discovery of patterns and relationships in huge databases and data mining is one of the core steps in the KDD process.

Knowledge discovery is the process of automatically generating information formalized in a form 'understandable' to humans. To bridge the gap of analyzing large volume of data and extracting valuable information and knowledge for decision making using new computerization technologies, DM and KDD has emerged since recent years.

According to U. Fayyad **[58]** KDD will continues to evolve, from the intersection of research in various fields like artificial intelligence , databases, machine learning, pattern recognition, statistics, knowledge acquisition for expert systems, data visualization, high-performance computing, machine discovery, scientific discovery and information retrieval. KDD software systems incorporate theories, algorithms, and methods from all of these fields.

Although, the two terms KDD and DM are closely related, yet they refer to slightly different two concepts. Data mining is only the application of a specific algorithm based on the overall goal of the KDD process. The knowledge discovery stage then extracts the knowledge which must then be post processed to facilitate human understanding. Post-processing usually takes the form of representing the discovered knowledge in a user friendly display.

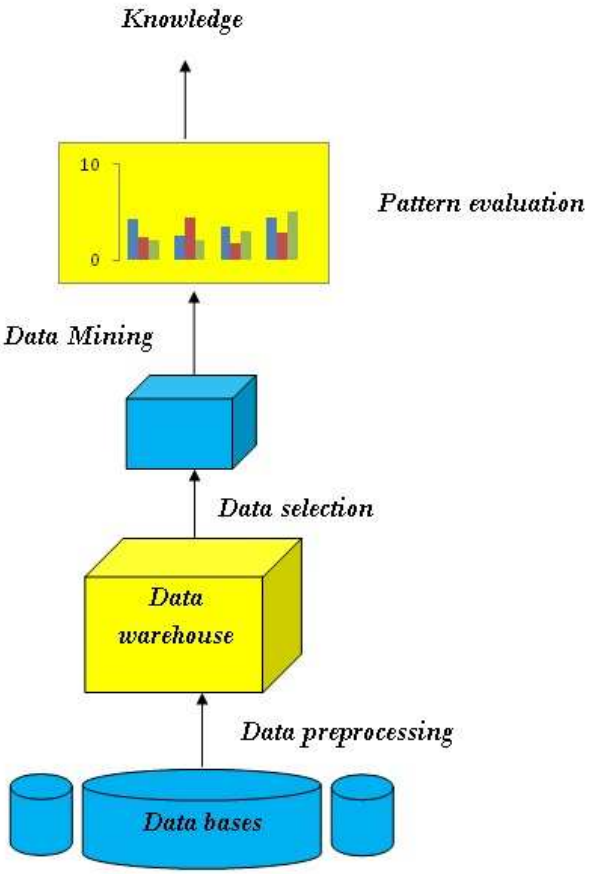


Figure 2.1 KDD process model

Data mining can mine data from different data storage [29][30] like text data, databases, data warehouse, transactional data, multimedia data , stream, spatiotemporal, time-series, sequence, and web, multi-media, graphs & social and information networks etc. The field of data mining grew out of the limitations of current data analysis techniques in handling challenges posed by these new types of datasets.

2.4.2. History of data mining.

The term "Data mining" was introduced in the 1990s, but data mining is the progress of a field with a long history [3] . Data mining roots are traced back along

three family lines: statistics , artificial intelligence [19], and machine learning [33] which are shown in Figure 2.2.

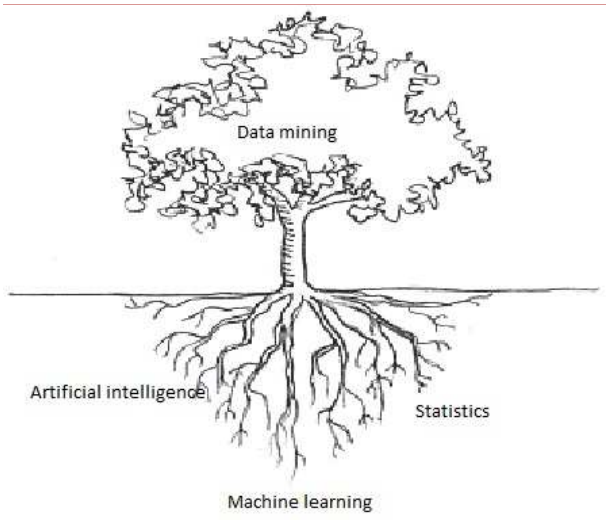


Figure 2.2 Data Mining and Associated Fields

Statistics is the foundation of many technologies on which data mining is built, e.g. regression analysis, standard distribution, standard deviation, standard variance, discriminant analysis, cluster analysis, and confidence intervals. All of these are used to study data and data relationships.

Artificial intelligence (AI), which is built upon heuristics as contrasting to statistics, it try to apply human-thought-like processing to statistical problems. Certain AI concepts which were adopted by some high-end commercial products, such as query optimization modules for Relational Database Management Systems .

Machine learning (ML) [5] is the combination of statistics and AI. It could be considered an evolution of AI, because it blends AI heuristics with advanced statistical analysis. Machine learning attempts to let computer programmes learn about the data they study, such that programmes make different decisions based on the qualities of the studied data, using statistics for fundamental concepts, and adding more advanced AI heuristics and algorithms to achieve its goals.

Data mining is adaptation of machine learning techniques to business applications. Data mining is best described as the union of historical and recent developments in statistics, AI, and ML. These techniques are then used together to study data and find patterns, rules and hidden trends.

In preliminary days, data mining algorithms mainly developed for numerical data but it further extended for all types of data like text, web, picture, multimedia spatial etc. as data mining began with analysis of single data base, but data mining techniques have evolved for flat files, traditional and relational databases and data warehouse. Later on, with the confluence of Statistics and Machine Learning techniques, various algorithms evolved to mine structured and unstructured data.

The field of data mining **[61]** has been greatly influenced by the development of fourth generation programming languages and various related computing techniques. In early days of data mining, most of the algorithms employed only statistical techniques. Later on, they evolved with various computing techniques like AI, ML and Pattern Reorganization. Various data mining techniques (Induction, Compression and Approximation) and algorithms developed to mine the large volumes of heterogeneous data stored in the data warehouses.

The field of data mining has been growing due to its enormous success in terms of scientific progress and broad-ranging application achievements and, understanding. Various data mining applications have been successfully implemented in various domains like financial analysis, customer management, health care, retail, telecommunication, fraud detection and risk analysis etc. The ever increasing complexities in various fields and improvements in technology have posed new challenges to data mining; the various challenges include different data formats, data from disparate locations, advances in computation and networking resources, research and scientific fields, ever growing business challenges etc.

2.4.3. Data mining functionality

Data mining is extraction of interesting patterns or knowledge from huge amount of data. For extraction of patterns various functionalities are available. Data

mining searches for non-trivial and implicit patterns from data. These patterns are mostly previously unknown but potentially useful. Data mining offers various types of functionalities, specific functionality is selected depending on the application area and kind of knowledge to be mined. Using these functionalities different type of knowledge can be mined like association rule, classification rule, discriminant rule and deviation analysis etc. Data mining functionalities [42] are extensive and rich; it can serve various fields and applications.

Figure 2.3 shows basic functionalities like classification, clustering, frequent pattern mining, outlier analysis etc. these functionalities are explained below.

Figure 2.3 Data mining functionalities

- **Characterization and Discrimination**

Data characterization [61] is a summarization of the general characteristics or features of a target class of data. In data characterization, based on users specific requirement summarization is done. The data is usually collected by a query. In data discrimination the target class data objects is compared with the objects from one or multiple contrasting classes with respect to specified generalized feature(s)[10][15]

- **Mining frequent patterns**

Frequent patterns [33] are the patterns that occur frequently in the data. Patterns can include itemsets, sequences and subsequences. A frequent itemset refers to a set of items that often appear together in a transactional data set.

Given a collection of items and a set of records, each of which contain some number of items from the given collection, an association function is an operation against this set of records which return, affinities or patterns that exist among the collection of items. These patterns can be expressed by rules such as "80% of all the records that contain items A, B and C also contain items D and E." The specific percentage of occurrences (in this case 80) is called the confidence factor of the rule. Also, in this rule, A,B and C are said to be on an opposite side of the rule to D and E. Associations can involve any number of items on either side of the rule.

- **Classification and prediction**

Classification [29] techniques in data mining are capable of processing a large amount of data. Classification assigns items in a data set to target categories or classes. Classification correctly predicts the target class for each case in the data. Classification consists of assigning a class label to a set of unclassified cases. Because the class label of each training tupple is provided, this step is also known as supervised learning also.

Classification techniques infer a model from the database. The database contains many attributes that denote the class of a tupple and these are known as predicted attributes whereas the remaining attributes are called predicting attributes. A combination of values for the predicted attributes defines a class.

When learning classification rules, the system has to find the rules that predict the class from the predicting attributes, so firstly the user has to define conditions for each class; the data mine system then constructs descriptions for the classes. Basically, the system should given a case or tupple with certain known attribute values be able to predict what class this case belongs to.

Once classes are defined the system should infer rules that govern the classification therefore the system should be able to find the description of each class. The descriptions should only refer to the predicting attributes of the training set so that the positive examples should satisfy the description and none of the negative. A rule said to be correct, if its description covers all the positive examples and none of the negative examples of a class.

There are various data mining classification techniques like Decision Tree based Methods, Rule-based Methods, Naïve Bayes and Bayesian Belief Networks, Nearest-Neighbor Method, Neural Networks, Support Vector Machines [25], Ensemble Methods usable for classification and prediction. Figure 2.4 shows classification using decision tree.

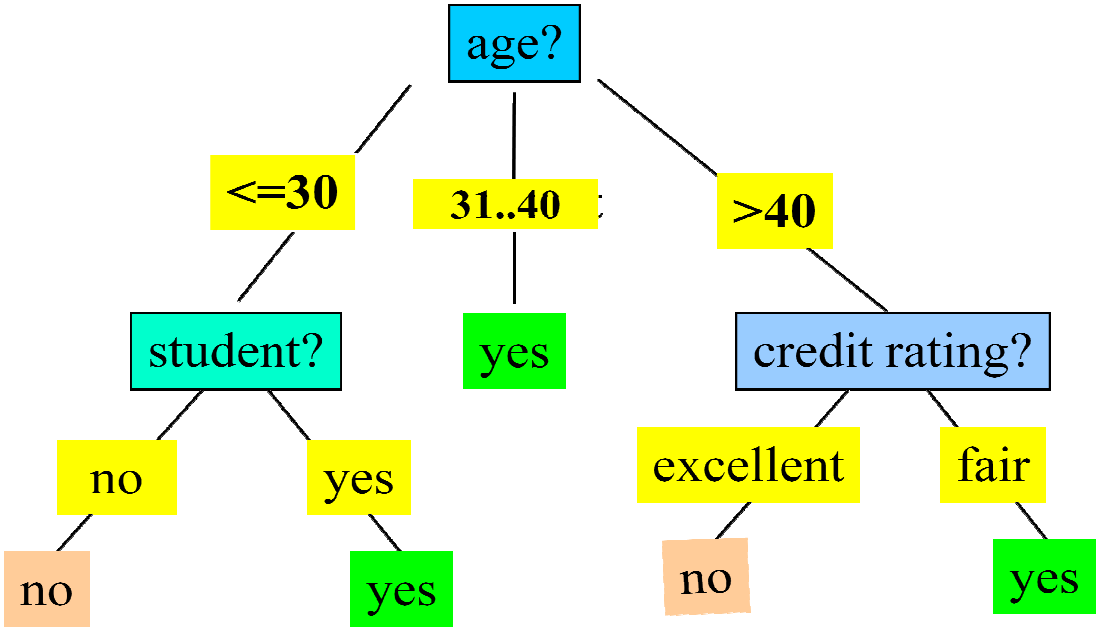


Figure 2.4 Classification using decision tree

- **Clustering**

Clustering [61] and segmentation are the processes of creating a partition so that all the members of each set of the partition are similar according to some metric. Clustering method belongs to unsupervised technique. In unsupervised technique classes or categories are not predefined. In this a set of objects grouped together

because of their similarity or proximity. When learning is unsupervised, the system has to discover its own classes i.e. the system clusters the data in the database. The system has to discover subsets of related objects in the training set and then it has to find descriptions that describe each of these subsets.

Objects are often decomposed into an exhaustive and/or mutually exclusive set of clusters.

Clustering [29] according to similarity is a very powerful technique, the key to it being to translate some intuitive measure of similarity into a quantitative measure. There are a number of approaches for forming clusters. One approach is to form rules which dictate membership in the same group based on the level of similarity between members. Another approach is to build set functions that measure some property of partitions as functions of some parameter of the partition. Figure 2.5 shows clustering data mining functionality.

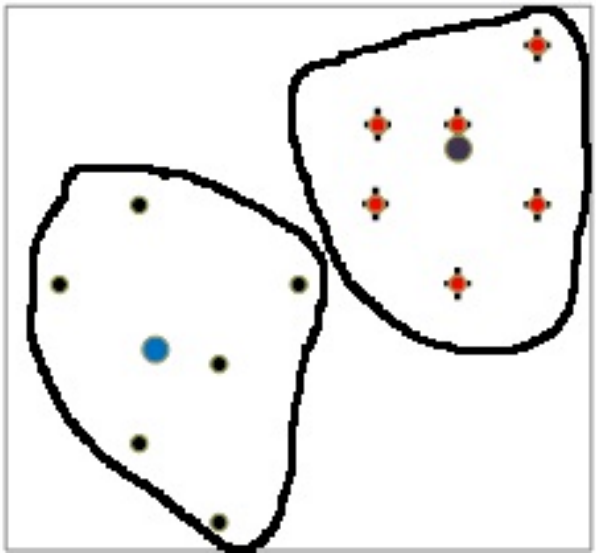


Figure 2.5 clustering.

- **Outlier analysis**

Outliers [29] are data objects that do not comply with the general behaviour or model of data. Outliers (if present in dataset) are discarded before processing through

other data mining functionalities. outliers usually represents exceptions or noise. Figure 2.6 shows outlier analysis, R represent data which is outlier from rest of data.

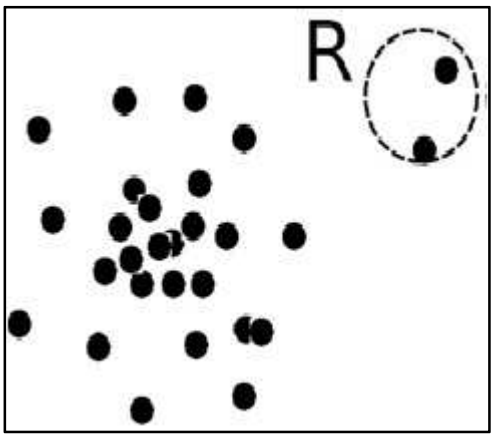


Figure 2.6 outlier analysis.

Data mining functionalities covers wide range of applications however there is need of new functionalities. Data mining research can provide new functionalities which can serve many application areas efficiently. Research in data mining has multiple aspects, if handled properly works effectively.

2.4.4. Data preprocessing

Before data is fed into a Data Mining algorithm, it must be collected, examined, cleaned and selected [28][29] . This entire process is called data preprocessing. The generation of raw data into machine understandable format is called preprocessing. If the data is of bad quality then even the best predictor will fail. Each algorithm requires data to be entered in a specified format.

Usually data is stored in formats like text, Excel or other database types of files. Generally free databases that are available online, the majority of them are in comma separated value (CSV) format. That is, all the attributes are separated by commas and missing data attribute is represented by two commas simultaneously. The majority of data mining tools can use data in the CSV format for running the machine intelligent algorithms. The data that is used for WEKA should be made into ARFF file format.

Sometimes, the raw data is not in any format. For better time efficiency with respect to processing of the data, algorithms need data in specific format. Therefore converting data in to specific format is very essential task.

- **Data cleaning**

There are a many data pre-processing [61] techniques available, data cleaning is one of them. To remove inconsistencies in the data, Data cleaning techniques are applied. These data processing techniques, when applied prior to mining, can significantly improve the overall data mining outcome. Data cleaning techniques clean the data by recognizing redundant or duplicate data and removing them. Data cleaning also resolve inconsistencies available in data. If the data is dirty, then results of data mining are not trustworthy. Furthermore, dirty data also causes confusion in the mining procedure, resulting in an unpredictable output. Therefore usefulness of data cleaning is significant. Although, many mining techniques have some procedures to deal with noisy or incomplete data, they are not always robust. Instead, they may concentrate on avoiding over fitting the data to the function being modeled. Therefore, data cleaning routines on data before data mining is must.

- **Replacing missing value**

If the data which is used for data mining have missing values it can give unpredictable results. There are many methods available to replace missing values. Following are the common methods.

- Manually fill the missing value.
- Using a global constant to fill in the missing values.
- Using the attribute mean to fill in the missing value.
- Removing the tupples having missing values.
- Using the most probable value to fill in the missing value.
- Running a clustering algorithm and replacing the missing attributes with the attributes of cases that appears close in an n-dimensional space.

The most common method of filling the attributes rapidly and without too much computation is to replace all the missing values with the arithmetic mean [36] or the mode with respect to that attribute.

- **Removing redundant and unnecessary attributes**

There is a possibility of redundant data in data set. Having a large amount of redundant data confuses the knowledge discovery process. It also slow down the work. Hence, the redundant data must be removed from data set .this process is usually done during data cleaning. To remove redundant data, the entire dataset is searched in sequential manner to test whether a tuple is redundant i.e. whether a tuple is repeated one or more times in the data set.

- **Feature selection[20][54]**

Depending on the required output only some attributes are required from the dataset. So irrelevant attributes need to be removed from the data to be mined. Only relevant features left after feature removal is presented as input to the data mining algorithm. It is observed that this analysis gives good classification rate and minimum error rate when compared to the classification done using the full feature set. Further, many data mining algorithms don't perform well with large amounts of features.tes. Therefore, feature selection techniques needs to be applied before data mining algorithm is applied. For feature selection filter method is used in this research work. Supervised attribute selection method is used before classification.

2.4.5. Classification methods

Classification [46] is identified as significant technique of data mining. Classification is a data mining function that assigns items in a dataset to target group or classes. Classification precisely predicts the target class for each item in the data.

A classification task begins with a data set in which the class assignments are predefined. For example, a classification model that predicts performance of student in exam is developed based on large amount of history data. In addition to the historical data, the data might track student's personal details and academic history.

In the model build (training) [47] process, a classification algorithm finds relationships between the values of the predictors and the values of the target. Every classification algorithms uses different techniques for finding relationships. These relationships are summarized in to a training model, then this training model is applied to a new data set in which the class assignments are unknown. Once model is trained effectively it precisely identifies the class of new data.

A classification model is tested by applying it to test data with known target values and comparing the predicted values with the previously known values. The test data must be compatible with the data used to train the model and must be preprocessed in the same way that the train data was prepared.

2.4.5.1 Decision tree

Decision trees [29][61] are the most useful tools for classification. Further, decision trees are used for prediction of classes. Decision trees generate rules which are easy to understand and usable in database access languages. In comparison to neural networks, decision trees rules are less complex. Decision tree also generate more accurate rules for data classification.

Decision tree is a classifier in the form of a tree structure, where each node is either: a leaf node or a decision node. Leaf node shows the value of the target class of data whereas decision node tests condition for a specific attribute and generate single or multiple branches depending on condition. A decision tree can be used to classify data by starting at the root of the tree and moving through it, until a leaf node, which provides the classification of the instance.

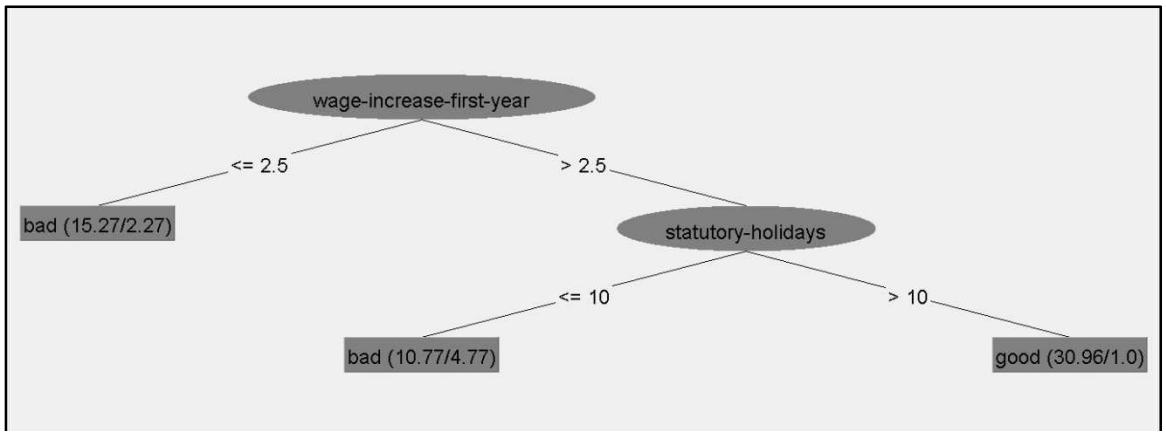


Figure 2.7 Decision Tree

Decision tree induction is a typical inductive approach to learn knowledge on classification. Or classification using decision tree following are required

- Training data: this data set have predefined classes for each instance of data . Usually it is called supervised.
- Large amount of data: Sufficiently large data is required for classification. Usually hundreds or thousands of training cases are required.
- All the instances available in dataset must belong to fixed collection of properties. This means that there is need to discretize continuous attributes.
- Discrete classes: A case does or does not belong to a particular class, and there must be more cases than classes.

Decision trees offer many advantages, some are mentioned below.

- It generates easy to understand rules. Rules are stored in the form of branches of tree.
- It can be applied to any type of data.
- Decision trees are easy to store and handle
- Handles very efficiently conditional information, it divide into sub branches and every branch is handled separately.

- The resulting trees are usually quite understandable and can be easily used to obtain a better understanding of the phenomenon in question. This is the most important of all the advantages listed.

The basic algorithm for decision tree is the greedy algorithm that constructs decision trees in a top-down manner with recursive divide-and-conquer approach.

The strengths of decision tree methods are:

- Decision trees are able to generate understandable rules.
- Decision trees perform classification with less computation.
- Decision trees are able to handle both continuous and discrete variables.
- Decision trees provide clear idea about which fields are most important for prediction or classification.

In some applications, the accuracy of a classification or prediction is the only thing that matters. There are a variety of algorithms for building decision trees that share the desirable quality of interpretability. A well known and frequently used over the years is C4.5.

Decision tree classification Algorithm [44]

Input

- D is training dataset with labels
- A is list of attribute.
- Feature selection method, a procedure which determines the splitting criteria for partitions of the data tupple into individual classes. This criterion consists of splitting attribute and split point .

Algorithm

Step 1. Create a node N of decision tree

Step 2. if all the tupples in D belong to the same class C then label N with class C

return N as leaf node.

Step 3. If A is empty then label N with Majority class C

Return N as leaf node

Step 4. Apply Feature selection method (D, A) to find the best splitting criterion

Label node N with splitting criteria;

Step 5. If splitting criteria is discrete valued and multiway split allowed then

A <- A -splitting attribute;

//remove splitting attribute.

Step 6. For each outcome j of splitting criterion

//partition tupple and grow subtree for each partition

Let Dj be the set of data tupples in d satisfying outcome j

If Dj is empty then

Attach a leaf label with majority class D to node N

Else attach the node returned by generate decision tree(Dj , A) to node N

End for

Step 7. Return N

Decision trees are construction starts by identifying the most useful attribute for classifying examples. Selection of attribute at every node is very important. This selection work is done with the help of statistical property *information gain.*

Information gain is a good quantitative measure of the worth of an attribute. It measures how properly a given attribute classify the training examples according to their target classification. This measure is used to select among the candidate attributes at each step while growing the tree.

Information gain [29] concept is based on entropy. It is a measure of homogeneity of instances. entropy is characterizes the impurity or purity of an arbitrary collection of examples. Given a set S, containing only positive and negative examples of some target concept , the entropy of set S relative to this simple, binary classification is defined as:

$$\text{Entropy}(S) = - p_p \log_2 p_p - p_n \log_2 p_n$$

where $p_p$is the proportion of positive examples in S and p_n is the proportion of negative examples in S. In all calculations involving entropy , 0log0 is defined to be 0.

The process of selecting a new attribute and partitioning the training examples is now repeated for each non-terminal descendant node, this time using only the training examples associated with that node. Attributes that have been incorporated higher in the tree are excluded, so that any given attribute can appear at most once along any path through the tree. This process continues for each new leaf node until either of two conditions is met:

1. Every attribute has already been included along this path through the tree.
2. The training examples associated with this leaf node all have the same target attribute value (i.e., their entropy is zero).

Algorithm for Attribute selection

Step 1. Compute entropy

$$\text{Entropy}(S) = \sum - p_i \log_2 p_i$$

//where p_i is the proportion of S belonging to class *i*.

Step 2. **Compute** information gain, *Gain (S, A)* of an attribute *A*, relative to a collection of examples S, is defined as

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

//where *Values(A)* is the set of all possible values for attribute *A*, and S_v is the subset of *S* for which attribute *A* has value *v*

// (i.e., S_v = {s € S | *A(s) = v*}). The first term in the equation for *Gain* is just the entropy of the original collection *S* and the second term is the expected value of the entropy after S is partitioned using attribute *A*. The expected entropy described by this second term is simply the sum of the entropies of each subset S_v, weighted by the fraction of examples $|S_v|/|S|$caused by knowing the value of attribute *A*

Step 3. Select attribute which has the highest gain.

J48 algorithm

The J48 algorithm derived from C4.5 Algorithm [44] is used for building the decision tree model. The training of the decision tree classification models of the experimentation is done by employing the 10-fold cross validation and the percentage split classification models. J48 is one of decision tree algorithm of data mining. J48 algorithm contains some parameters that can be changed to further improve classification accuracy. Initially the classification model is built with the default parameter values of the J48 algorithm.

The J48 algorithm gives several options related to tree pruning. Pruning produces fewer, more easily interpreted results. More importantly, pruning can be used as a tool to correct for potential over fitting. The J48 algorithm recursively

classifies until the data has been classified as close to perfectly as possible. Pruning always reduces the accuracy of a model on training data.

J48 uses two pruning methods. The first is known as subtree replacement and second is subtree raising. In subtree replacement method, nodes in a decision tree are replaced with a leaf node it reduces the number of tests along a certain branch of treep. This process starts from the leaves of the fully formed tree, and works backwards toward the root. In subtree raising method ,a node moves upwards towards the root of the tree, it replaces other nodes along the way. Subtree raising method is computationally complex than subtree replacement.

Tree pruning calculates error rates to decide about which parts of the tree to replace or raise. This can be done in multiple ways. The simplest way is to reserve a portion of the training data to test on the decision tree, which helps to overcome potential overfitting, this approach is called reduced-error pruning. This method reduces the overall amount of data available for training the model. This approach is applied on large datasets whereas it advisable to avoid on small datasets.

Other error rate methods statistically analyze the training data and estimate the amount of error inherent in it. This is complex method, but forecast the natural variance of the data. This approach requires a confidence threshold, which by default is set to 25 percent. This option is important for determining how specific or general the model should be. If the training data is expected to conform fairly closely to the data, you'd like to test the model on, this figure can be lowered. The reverse is true if the model performs poorly on new data; try decreasing the rate in order to produce a more pruned tree.

There are several other options that determine the specificity of the model. The minimum number of instances per leaf is one powerful option. This allows you to dictate the lowest number of instances that can constitute a leaf. Higher the number more general the tree is. Lowering the number will produce more specific trees, as the leaves become more granular. The binary split option is used with numerical data. If turned on, this option will take any numeric attribute and split it into two ranges using an inequality. This greatly limits the number of possible decision points. Rather than

allowing for multiple splits based on numeric ranges, this option effectively treats the data as a nominal value. Turning this encourages more generalized trees. There is also an option available for using Laplace smoothing for predicted probabilities. Laplace smoothing is used to prevent probabilities from ever being calculated as zero. This is mainly to avoid possible complications that can arise from zero probabilities.

The most basic parameter is the tree pruning option. These options define the performance of classifiers. It is important to experiment with models by wisely adjusting these parameters. Often, only repeated experiments and familiarity with the data will give out the best set of options.

Options available for decision tree classifires

- binarySplits – this option Whether to use binary splits on nominal attributes when building the trees.
- Confidence Factor -- The confidence factor used for pruning (smaller values gives more pruning).
- Debug -- this option provides additional info to the console.
- minNumObj – This option defines minimum number of instances per leaf.
- numFolds -- Determines the amount of data used for reduced-error pruning. One fold is used for pruning, the rest for growing the tree.
- reducedErrorPruning -- Whether reduced-error pruning is used instead of C.4.5 pruning.
- saveInstanceData -- Whether to save the training data for visualization.
- seed -- The seed used for randomizing the data when reduced-error pruning is used.
- subtreeRaising -- Whether to consider the subtree raising operation when pruning.

- unpruned -- this option defines whether to prune tree or not.
- useLaplace – this option defines whether counts at leaves are smoothed based on Laplace.

2.4.5.2 Bayesian classifier

A Bayesian classifier [29] is based on the idea that the class can predict the values of features for members of that class. Instances are grouped in classes because they have common values for the features. These classes are called natural kinds. In this section, the target feature corresponds to a discrete class, which is not necessarily binary.

The idea behind a Bayesian classifier is that, if an agent knows the class, it can predict the values of the other features. If it does not know the class, Bayes' rule can be used to predict the class given (some of) the feature values. In a Bayesian classifier, the learning agent builds a probabilistic model of the features and uses that model to predict the classification of a new example.

Bayes net

Bayes Nets [61] or Bayesian networks are graphical representation for probabilistic relationships among a set of random variables. Given a finite set $S=\{S_1,\ldots S_n)$ of discrete random variables where each variable S_i may take values from a finite set, denoted by $Val(S_i)$. A Bayesian network is an annotated directed acyclic graph (DAG) G that encodes a joint probability distribution over S. The nodes of the graph correspond to the random variables $S_1,\ldots S_n$.

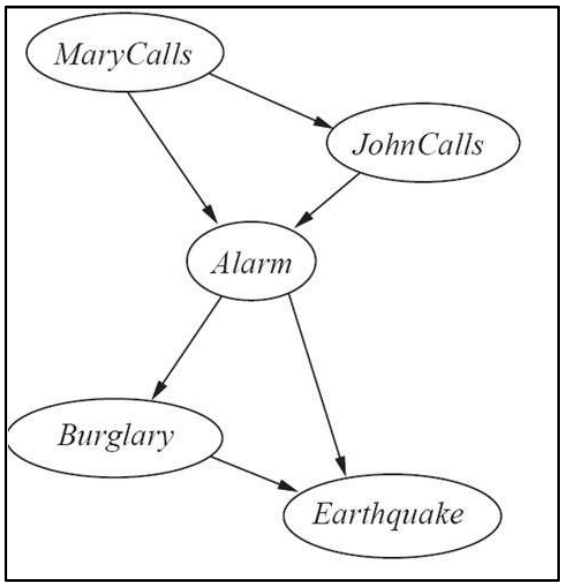


Figure 2.8 Bayesian classification

2.4.5.3 Rule based classifier

Rule based classifiers [29] use rule induction method. Rule induction methods identify and defines pattern available in dataset. In this all possible patterns are methodically pulled out of the data and then an accuracy and significance are added to them that tell the user how strong the pattern is and how likely it is to occur again. In rule induction systems, the rule itself is of a simple form of "if this and this and this then this". Rules are mutually exclusive and exhaustive.

One Rule algorithm of rule induction uses a greedy depth-first policy to identify patterns. Each time it is faced with adding a new attribute test to the current rule, it picks the one that most improves the rule quality, based on the training samples. OneR algorithm steps as are follows:

- Sequentially, learn one rule at a time,.
- After a rule is learned, the training instances covered by the rule are removed.
- Only the remaining data is used to find subsequent rules in the dataset.

The process repeats until some stopping criteria are met.

IF *age* = young AND *student* = *no* THEN *buys_computer* = *no*
IF *age* = young AND *student* = *yes* THEN *buys_computer* = *yes*
IF *age* = mid-age THEN *buys_computer* = *yes*
IF *age* = old AND *credit_rating* = *excellent* THEN *buys_computer* = *no*
IF *age* = old AND *credit_rating* = *fair* THEN *buys_computer* = *yes*

Figure 2.9 rule based classification

2.4.6. Ensemble methods

Ensemble Methods [17] [38] are based on the concept that multiple models can be used to train dataset. An ensemble classifier is a method which uses or combines multiple classifiers to improve robustness as well as to achieve an improved classification performance from any of the constituent classifiers. Furthermore, this technique is more flexible to noise compared to the use of a single classifier.

Ensemble learning methods instead generate multiple models. Given a new example, the ensemble passes it to each of its multiple base models.

- **Bagging**

  Bagging is one of most useful ensemble method. Bagging (Bootstrap Aggregating) generates multiple bootstrap training sets from the original training set and employs each of them to generate a classifier for inclusion in the ensemble. This method is usually applied to decision tree algorithms, but it also can be used with other classification algorithms such as naïve bayes, nearest neighbour, rule induction, etc. The bagging technique is very useful for large and high-dimensional data, such as intrusion data sets, where finding a good model or classifier that can work in one step is impossible because of the complexity and scale of the problem.

- **Boosting**

  Boosting is a forward stage wise additive model .Boosting, is an ensemble method for boosting the performance of a set of weak classifiers into a strong classifier. This technique can be viewed as a model averaging method and it was originally designed for classification, but it can also be applied to regression. Boosting provides sequential learning of the predictors. The first one learns from the whole data set, while the following learns from training sets based on the performance of the previous one. The misclassified examples are marked and their weights increased so they will have a higher probability of appearing in the training set of the next predictor. It results in different machines being specialized in predicting different areas of the dataset.

  - Adaboost

  AdaBoost algorithm which is one of the most widely used boosting techniques for constructing a strong classifier as a linear combination of weak classifiers. AdaBoost generates a sequence of base models with different weight distributions over the training set.

2.5. Supervised vs unsupervised learning methods

Data mining learning algorithms can be categorized into supervised [45] or 'unsupervised' [46]. This bifurcation depends on how the learner classifies data. basic requirement of supervised learning algorithm is predefined classes and availability of learning data. In supervised algorithms, mathematical model is constructed based on the patterns available in data set. These patterns are observed for predetermined classes. Data is labeled with these classifications. These models then are evaluated on the basis of their predictive capacity in relation to measures of variance in the data itself. Supervised learning algorithms are used in decision tree, bayes net etc.

Unsupervised learners are not provided with classifications. In fact, the basic task of unsupervised learning is to develop category wise labels automatically. Unsupervised algorithms seek out similarity between pieces of data in order to determine whether they can be characterized as forming a group. These groups are termed clusters, and there is a whole family of clustering machine learning techniques.

In unsupervised classification, often known as 'cluster analysis' the machine is not told how the texts are grouped. Its task is to arrive at some grouping of the data. In clustering initially criteria is provided for cluster construction. These criteria depends on density , number of partitions, hierarchy etc.

Table 2.1 Comparison of supervised and unsupervised learning

Name of learning method	Associated data mining functionality	Features	Popular Algorithms
Supervised	Classification	Class labells are known in advance	Decision tree, Bayesian methods, rule based classifiers.
Unsupervised	Clustering	Class labells are not known in advance	Partition based clustering method Kmean etc.

Table 2.1 shows supervised and unsupervised learning methods with Associated data mining functionality, Features and Popular Algorithms.

2.6. Data mining types of models

Data mining model are basically of two types as per usage i.e. descriptive model and predictive model.

Descriptive Model

Descriptive data mining [61] is normally used to generate correlation, frequency and cross tabulation. Descriptive method can be defined to discover interesting regularities in the data, to find previously unknown patterns and find interesting subgroups in the bulk of data. Under descriptive model clustering, association rules are used.

Predictive models

The goal of the predictive models [29] [61] is to construct a model by using the results of the known data and is to predict the results of unknown data sets. This work is done by using the constructed model. For instance, a bank might have the necessary data about the loans given in the previous terms. In this data, independent variables are the characteristics of the loan granted clients and the dependent variable is whether the loan is paid back or not. The model constructed by this data is used in the prediction of whether the loan will be paid back by client in the next loan applications. For predictive data mining classification and regression functionalities are used.

2.7. IDS (Intrusion detection system) Product review

"The global network security market could hit $9.5 billion by 2015" this is according to a report published by Global Industry Analysts. This report also mentions that Asia Pacific region is more receptive to network growth. Additionally, the Intrusion Detection System/Intrusion Prevention System (IDS/IPS) market is expected to become the second largest product segment of the network security market. IDS/IPS solutions will be in high demand because of their efficient methods to deal with cyber attacks. .

There are many factors which play important role while selecting IDS .

- **Product popularity**
    - Vendor capability Assess the vendor's technology and strategic viability.
    - Installation and Operating System Setup of a programme or software.
    - Automated Actions: Ability to automate specific actions and tasks to deliver operational efficiency.
- **Capacity to detect new intrusion**

  IDS should not only identify known attack but it must have capacity to identify new attacks.
- **Best user interface**
    - Ease of Use: It is the simplicity of the IDS tool to utilize and efficiently manage.
    - Advanced Displays: It is the ability to display various angles on attack data and correlated events.
    - Tagging: It is the new system that allows default or custom tags to be added to events.
    - User Log: It is the ability to monitor, manage and provide detailed logs of user activities.
    - Reporting Tools: user interface to put security data in an easy to understand format.
- **Accuracy of intrusion detection.**
    - Identifying areas of high risk: if areas of security threats are identified through IDS than that IDS gives better security solutions

- False Positive Protection: Ability to efficiently validate security events and identify potential false positives.
- Event integration: if multiple events are use to detect suspicious behaviour and network vulnerabilities.
- Live and Real-Time Monitoring: It is the ability to view and respond events in real-time.
- Scalability and Implementation: It is the ability to handle every network environment.
- Event Details: Ability to provide consistent and detailed security event information.

- **Other factors**
  - Forensic Analysis: Analysis of detected security events in the interest of figuring out what happened, when it happened, how it happened, and who was involved.

Presently, there are around many intrusion detection systems available in the network security market. Some of the popular IDS products are discussed below.

SNORT [50]

Snort is a platform independent, lightweight network intrusion detection tool that can be deployed to monitor small TCP/IP networks and detect a wide variety of suspicious network traffic as well as outright attacks. It is 'lightweight' because it can easily be deployed on almost any node of a network; it has a small footprint and can easily be configured by system administrators.

Snort is a packet sniffer based on libpcap and belongs to network intrusion detection system (NIDS) category of IDS. It features rules based logging and has real-time alerting capability. It can detect a variety of attacks by using concept of content

pattern matching. The detection engine is programed using a simple language that describes per packet tests and actions. Ease of use simplifies and accelerates the development of new exploit detection rules. It effectively identifies probes like SMB probes, buffer overflows, stealth port scans and CGI attacks, etc.

Snort's architecture is focused on performance, simplicity, and flexibility. There are three primary part of Snort:

- Packet decoder
- Detection engine
- Logging and alerting subsystem.

These subsystems ride on top of the libpcap promiscuous packet sniffing library, which provides a portable packet sniffing and filtering capability.

- **Dragon [13]**

Dragon is a family of IDS products from Enterasys Networks which includes Dragon Sensor, a network based intrusion detection system (NIDS); Dragon Squires, a host based intrusion detection system (HIDS) .

Dragon Sensor provides high-Bandwidth Support with correct tuning and architecture. It decodes the majority of frequently encoded protocols, reassembles UDP and TCP streams to disable attacks. It works beyond signature detection and provides anomaly-Based Detection . it detects buffer overflows and traffic profiling etc by using anomaly detection capacities. It provides two interfaces, first one to monitor network and second for reporting purpose.

Dragon Squire supports the majority of commercial firewalls. Firewall forward log to Dragon Squire System. It detects attacks directed particularly at Apache , Netscape web servers and IIS. it detects attacks directed at highly vulnerable applications ,it identifies frequently attacked applications which includes DNS servers, mail servers, FTP servers.

The Dragon sensor and squire are platform independent as they run on most of popular operating systems like Windows NT/2000, Linux ,Solaris, HP-UX, ,OpenBSD and FreeBSD via software license.

Other leading intrusion detection systems are as following:

• CounterAct [8]

CounterACT Edge security appliance offers an unique approach for intrusion prevention. It neither directly perform anomaly detection or signature detection but work on 'proven intent' of attackers. Attackers follow a consistent pattern. To launch an attack, they need knowledge about a network's resources. Usually system vulnerability and configuration are scanned or probed before attack. This is observed in most of human intruders or self-propagating threats. The information received is then used to launch attacks based on the unique structure and characteristics of the targeted network.

• Enterprise network security- Airmagnet [16]

AirMagnet Enterprise provides a simple and scalable Wireless network solution. All types of wireless threats are handled efficiently. Its tools ensure to accommodate highest capacity of network users and perform optimally. It provides proactive alerting and detailed report. It offers quick and effective troubleshooting and resolves almost all types of wireless issues. It has intrusion prevention system as well s intrusion detection system.

• Bro Intrusion Detection System [4]

Bro is an open-source Network Intrusion Detection System (NIDS), it is unix based. It submissively monitors network traffic and looks for suspicious activity. Its analysis includes detection of exact attacks and unusual activities. Bro at first parse network traffic and extract its application level semantics and then compare the activity with patterns deemed troublesome. Bro analyses real time as well as offline network data. Bro engine is written into C++.

• Cisco Intrusion Prevention System (IPS) [7]

Cisco IPS is one of the most widely deployed intrusion prevention systems, providing: Protection against more than 30,000 known threats, Timely signature updates and Cisco Global Correlation to dynamically recognize, evaluate, and stop emerging Internet threats. Cisco IPS includes industry-leading research and the expertise of Cisco Security Intelligence Operations. Cisco IPS also helps organization comply with government regulations and consumer privacy laws.

• Juniper Networks Intrusion Detection & Prevention (IDP) [31]

Juniper Networks IDP Series Intrusion Detection and Prevention Appliances with Multi-Method Detection, offers comprehensive coverage by leveraging multiple detection mechanisms. For example, by utilizing signatures, as well as other detection methods including protocol anomaly traffic anomaly detection, the Juniper Networks IDP Series appliances can thwart known attacks as well as possible future variations of the attack. Backed by Juniper Networks Security Lab, signatures for detection of new attacks are generated on a daily basis.

• McAfee Host Intrusion Prevention for server [39]

Defend servers from known and new zero-day attacks with McAfee Host Intrusion Prevention. Boost security, lower costs by reducing the frequency and urgency of patching, and simplify compliance.

• Sourcefire Intrusion Prevention System (IPS) [51]

Built on the foundation of the award-winning Snort® rules-based detection engine, Sourcefire IPS™ (Intrusion Prevention System) uses a powerful combination of vulnerability- and anomaly-based inspection methods—at throughputs up to 10 Gbps—to analyze network traffic and prevent critical threats from damaging network. Whether deployed at the perimeter, in the DMZ, in the core, or at critical network segments, and whether placed in inline or passive mode, Sourcefire’s easy-to-use IPS appliances provide comprehensive threat protection.

• Strata Guard IDS/IPS [53]

The award-winning Strata Guard high-speed intrusion detection/prevention system (IDS/IPS) gives you real-time, zero-day protection from network attacks and malicious traffic, preventing Malware, spyware, port scans, viruses, and DoS and DDoS from compromising hosts, Device and network outages, Data leakage, High-risk protocols, such as BitTorrent, Kazaa, and TelNet, from running on network, Unauthorized access to sensitive data.

2.8. Summary and evaluation of existing IDS products

Current IDSs generate too many inaccurate alarms. Simply stated, IDSs aren't good enough yet. There are many factors to consider when evaluating IDSs such as speed, cost, effectiveness, ease-of-use, scalability, and interoperability. Without taking specific environment details into consideration, effectiveness and ease-of-use can be used as general metrics to compare IDSs. Both factors measure general aptitude because they are determined by the detection algorithm of the IDS.

The detection algorithm maps incoming events to attacks and normal activity. The resulting classification can be used to determine the effectiveness of IDS. Effectiveness is the ability of an IDS to maximize the detection rate while minimizing the false alarm rate (false positive rate). In other words, good IDS reports intrusions when they occur, and does not report intrusions when they do not occur. The probability that an intrusion is actually occurring, given that an IDS reports an intrusion, is dominated by the false alarm rate of the IDS. The important measure of an IDS is not how frequently it detects attacks, but how infrequently it produces false alarms.

Another important factor for measuring IDSs is its ease-of-use. Because active response is not yet an acceptable technology, human intervention is necessary to use IDSs. It is therefore necessary for IDSs to be intuitive and easy to manage.

Drawbacks of current IDS:

Current IDS (Intrusion Detection Systems) have from two major drawbacks.

1. False positive generation.

   Common complaint is the amount of false positive. False positive means if the network packet is normal then also IDS gives alarm of attack. If false positive comes frequently then if attack exists then also it is not taken seriously.

2. False negative generation.

   False negative [14] means attack is not accurately identified and it is considered normal. In this case attacks which are bypassed through IDS, can cause severe harm to computer and do not solve actual purpose of intrusion detection.

IDS is one of standard component in security infrastructures, it allow network administrators to detect intrusion. Intrusion attacks may include internal attack (or misuse) and external attacks.

2.9. Summary and evaluation of Literature Review

Based on the reviewed literature, we can conclude that majority of researchers has given evidence of successful implementation of IDS using data mining techniques. In particular, some of the work on intrusion detection, use of data mining techniques for intrusion detection ,challenges faced by the intrusion detection domain are highlighted.

- Various methods are used by researchers like association rules mining, Density based outlier detection scheme, clustering, ensemble methods, genetic algorithm, neural network, classification methods. table 2.2 presents summary and comparison of literature review based on data mining methods for IDS.

Table 2.2 Comparison of literature review based on data mining methods for IDS

Sr. No.	Author	Title	Year	Technique	Features
1.	Wenke Lee	“A Data Mining Framework for Building Intrusion Detection Models ”	1996	Multiple base classifiers can be	Misuse detection system

				combined	
2.	Daniel Barbara et al.	"ADAM: Detecting Intrusions by Data Mining"	2001	Pseudo-Bayes estimators.	Anomaly detection system
3.	Levent Ertoz, Eric et al.	"The MINDS – Minnesota Intrusion Detection System"	2004	Density based outlier detection	Unsupervised Anomaly detection
4.	Fangfei Weng	"An Intrusion Detection System Based on the Clustering Ensemble"	2007	Clustering Ensemble,	Unsupervised anomaly detection
5.	Herkshop S.	"A data mining approach to host based intrusion detection"	2007	Adaptive learning algorithms	Unsupervised anomaly detection algorithms
6.	T. Lappas et al.	"Data Mining Techniques for (Network) Intrusion Detection System"	2007	Bi-clustering	Unsupervised anomaly detection algorithms
7.	Kamran Shafi	"An Online and Adaptive Signature-based Approach for Intrusion Detection Using Learning Classifier Systems"	2008	Adaptively learning maximally rules	Signature extraction
8.	Flora S. Tsai	Network Intrusion Detection Using Association Rules	2009	Association Rules	
9.	Dewan Md. Farid et al.	"Attacks Classification in Adaptive Intrusion Detection using Decision Tree"	2010	Adaptive Intrusion Detection using	Supervised anomaly detection

				Decision Tree	
10.	Ghanshyam Prasad Dube et al.	" A Novel Approach to Intrusion Detection System using Rough Set Theory and Incremental SVM"	2011	RST (Rough Set Theory) and Incremental SVM (Support Vector Machine)	Unsupervised
11.	Huu Hoa Nguyen et al.	"An Efficient Fuzzy Clustering-Based Approach for Intrusion Detection"	2011	Fuzzy Clustering-Based Approach	Unsupervised
12.	Gunja Ambica et al.	"Robust Data Clustering Algorithms for Network Intrusion Detection"	2012	Clustering	Unsupervised anomaly detection
13.	Nagaraju Devarakonda et al.	"Intrusion Detection System using Bayesian Network and Hidden Markov Model"	2012	HMM and Bayesian Network	Supervised anomaly detection
14.	Shyara Taruna et al.	"Enhanced Naïve Bayes Algorithm for Intrusion Detection in Data Mining"	2013	Naïve bayes	Supervised Anomaly detection
15.	G.V. Nadiammai et al.	"Effective approach toward Intrusion Detection System using data mining techniques"	2014	Hybrid PSO	Semisupervised

- Most of the literature reviewed revealed that the earlier studies were mainly related to misuse detection model whereas current studies are related to anomaly detection. Misuse detection model is also known as signature analysis. The strength of signature analysis depends upon the quality, comprehensiveness, and timeliness of the attack signature housed in the IDS's search engine. However, despite the variety of such methods described in the literature in recent years, security tools incorporating anomaly detection functionalities are just starting to appear, and several important problems remain to be solved.

  Many researchers worked on anomaly detection model. Anomaly-based detectors attempt to estimate the ''normal'' behaviour of the system to be protected and generate an anomaly alarm whenever the deviation between a given observation at an instant and the normal behaviour exceeds a predefined threshold.

- Low detection efficiency, especially due to the high false positive rate usually obtained This feature is generally explained as arising from the lack of good studies on the nature of the intrusion events. The problem calls for the exploration and development of new, accurate processing schemes, as well as better structured approaches to modelling network systems.

- The researchers studied work done in the related area and concluded that data mining based two types of approaches are used for construction of intrusion detection ,first approach is supervised approach and other is unsupervised approach. Researcher has discovered that supervised approach works better than unsupervised approach.

2.10. Chapter summary

In this chapter summary of the information collected from various sources in the form of secondary data is available. The information is collected from reference books, research papers, technical white papers, journals and web sites. This chapter provides background of the earlier studies in the similar subject. A review of the work in the intrusion detection domain related to this research's approach is presented.

2.11. Chapter References

1. Adeyinka, O., (2008), "Internet Attack Methods and Internet Security Technology Modeling & Simulation" ,AICMS 08. Second Asia International Conference on,vol., no., pp.77-82.

2. Bishwanath mukharjee L.Todd heberlien, (1994), "Network Intrusion Detection" ,IEEE.

3. Bhavya Daya , (2010), "Network Security: History, Importance, and Future", University of Florida Department of Electrical and Computer Engineering.

4. Bro network security monitor , www.bro.org, Accessed date Jan 2012

5. C. F. Tsai, Y. F. Hsu, C. Y. Lin and W. Y. Lin, (2009), " Intrusion detection by machine learning: A review" ,, Expert Systems with Applications, Vol 36, Issue 10, pp. 11994-12000. 2009.

6. Chia-Mei Chen , Ya-Lin Chen, Hsiao-Chung Lin,(2010) "An efficient network intrusion detection" ,Computer Communications 33 ,477–484.

7. Cisco intrusion detection , www.cisco.com, Accessed date Jan 2012

8. Counteract edge for threat prevention www.forescout.com/product /counteract-edge, Accessed date Jan 2012

9. Daniel Barbara, Julia C., (2001),"ADAM: Detecting Intrusions by Data Mining" , Proceedings of the 2001 IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY, 5.

10. Dash M. and Liu H.,(1997), "Feature selection for classification", Intelligent Data Analysis: An International Journal, PP. 131–156.

11. Dewan Md. Farid, Nouria Harbi, Emna Bahri, Mohammad Zahidur Rahman, Chowdhury Mofizur Rahman ,(1010),"Attacks Classification in Adaptive Intrusion Detection using Decision Tree", World Academy of Science, Engineering and Technology 63.

12. Dorothy E. Denning , (1987),“An Intrusion-Detection Model”, IEEE Transactions On Software Engineering, Vol. Se-13, No. 2, , 222-232. .

13. Dragon documentation http://www.nuance.com/naturallyspeaking/resources/ documents/ usergd_v11.pdf , Accessed date Jan 2012

14. E.Kesavulu Reddy, Member IAENG, V.Naveen Reddy, P.Govinda Rajulu,(2011), “ A Study of Intrusion Detection in Data Mining ” ,Proceedings of the World Congress on Engineering 2011 Vol III WCE 2011, , London, U.K. ISSN: 2078-0966 (Online).

15. Ellen Pitt and Richi Nayak, (2007),“The Use of Various Data Mining and Feature Selection Methods in the Analysis of a Population Survey Dataset”,Conferences in Research and Practice in Information Technology.

16. Enterprise network security- airmagnet |flukenetworks www.flukenetworks.com , Accessed date Jan 2012

17. Fangfei Weng, Qingshan Jiang, Liang Shi, and Nannan Wu,(2007), “An Intrusion Detection System Based on the Clustering Ensemble”, IEEE.

18. Flora S. Tsai ,(2009), “Network Intrusion Detection using Association Rules” International Journal of Recent Trends in Engineering, Vol 2, No. 2.

19. G. Kumar, K. Kumar and M. Sachdeva, (2010),s“ The Use of Artificial Intelligence based Techniques For Intrusion Detection – A Review, Artificial Intelligence Review” , vol. 34, No. 4, pp. 369-387, Springer, Netherlands, DOI: 10.1007/s10462-010-9179-5 ISSN: 0269-2821.

20. G. Kumar, K. Kumar, M. Sachdeva,(2010), “ An Empirical Comparative Analysis of Feature Reduction Methods For Intrusion Detection” , International Journal of Information and Telecommunication, 1 , 44-51, ISSN: 0976-5972.

21. G.V. Nadiammai, M. Hemalatha ,(2014),"Effective approach toward Intrusion Detection System using data mining techniques", Cairo University,Egyptian Informatics Journal,www.elsevier.com/locate/eij ,1110-8665 .

22. Ghanshyam Prasad Dubey, Prof. Neetesh Gupta, Rakesh K Bhujade,(2011), " A Novel Approach to Intrusion Detection System using Rough Set Theory and Incremental SVM" ,International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-1.

23. Gregory Piatetsky-Shapiro, Christopher Matheus, Padhraic, Smyth, and R amasamy Uthurusamy,(1994), " KDD-93: Progress and Challenges in Knowledge Discovery in Databases"

24. Gunja Ambica et al.(2012) ," Robust Data Clustering Algorithms for Network Intrusion Detection" , International Journal of Computer & Organization Trends –Volume2Issue5- 2012 ISSN: 2249-2593 Page 118

25. Hershkop S., Apap F., Eli G., Tania D., Eskin E., Stolfo S., (2007),"A data mining approach to host based intrusion detection" , Technical reports, CUCS Technical Report.

26. Hulus onder, (2007),"A security management system design" , thesis , middle east technical university.

27. Huu Hoa Nguyen, Nouria Harbi and Jerome Darmont, (2011)"An Efficient Fuzzy Clustering-Based Approach for Intrusion Detection"

28. Ilia Mitov, Krassimira Ivanova, Krassimir Markov,Vitalii Velychko, Peter Stanchev, Koen Vanhoof, (2008), "comparison of discretization methods for preprocessing data for pyramidal growing network classification method" , International Book Series "Information Science and Computing".

29. Jiawei Han And Micheline Kamber,(2008), "Data mining concepts and techniques" , Morgan Kaufmann publishers .an imprint of Elsevier .ISBN 978-1-55860-901-3. Indian reprint ISBN 978-81-312- 0535-8 .

30. Joyce Jackson, (2002), "Data Mining: A Conceptual Overview" Communications of the Association for Information Systems ,Volume 8.

31. Juniper network intrusion detection www.juniper.net, Accessed date Jan 2012

32. Kamran Shafi, (2008),"An Online and Adaptive Signature-based Approach for Intrusion Detection Using Learning Classifier Systems" ,THESIS, University of New South Wales.

33. Kayacik, G. H., Zincir-Heywood, A. N.,(2005), "Analysis of Three Intrusion Detection System Benchmark Datasets Using Machine Learning Algorithms" , Proceedings of the IEEE ISI 2005 Atlanta, USA.

34. KdNuggets,(2007), "Data Mining Methodology", http://www.kdnuggets.com /polls/2007/ datamining_methodology.htm,

35. Kingsly Leung Christopher Leckie y.,(2005), "Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters" ,28th Australasian Computer Science Conference, The University of Newcastle, Australia,. Conferences in Research and Practice in Information Technology, Vol. 38.

36. Laura Ruotsalainen,(2008), Data Mining Tools for Technology and Competitive Intelligence,VIT, ISBN 978-951-38-7240-3.

37. Levent Ertoz, Eric Eilertson, Aleksandar Lazarevicy, Pang-Ning Tan, Vipin Kumar, Jaideep Srivastava_y, Paul Dokas., (2004), "The MINDS – Minnesota Intrusion Detection System" .

38. M.Govindarajan and RM.Chandrasekaran, (2012) ,"Intrusion Detection using an Ensemble of Classification Methods" ,Proceedings of the World Congress on Engineering and Computer Science 2012 Vol I WCECS 2012, , San Francisco, USA..

39. Mcafee network intrusion prevention, www.mcafee.com ,Accessed date Jan 2012

40. Meenakshi.RM, Mr.E.Saravanan,(2013), “A data mining analysis & approach with intrusion detection / prevention from real”.

41. Michael J. Pazzani , (2000), “Knowledge discovery from DATA?”, IEEE Intelligent Systems.

42. Mohammadreza Ektefa , Sara Memar, Fatimah Sidi, Lilly Suriani Affendey.,(2010),“Intrusion Detection Using Data Mining Techniques ” , 978-1-4244-5651-2/10/2010.

43. Nagaraju Devarakonda et al. “Intrusion Detection System using Bayesian Network and Hidden Markov Model”, Available online at www.sciencedirect.com, elsevier , Procedia Technology 4 (2012) 506 – 514

44. Quinlan, J.R. ,(1993), “C4.5: Programs For Machine Learning”, San Mateo, CA: Morgan Kaufmann.

45. Rajashree Dash, Rajib Lochan Paramguru, Rasmita Dash,(2011) “Comparative Analysis of Supervised and Unsupervised Discretization Techniques”.

46. Rajni jain ,(2011), “Introduction To Datamining Techniques” ,www.iasri.res.in/ebook/ expertsystem/DataMining.pdfSimilar

47. Ramageri B. M. ,(2011), “Data Mining Techniques and Applications” , In Indian Journal of Computer Science and Engineering Vol. 1 No. 4 pp 301-305,2011.

48. Roman V. Yampolskiy and Venu Govindaraju,(2007), “Computer Security: a Survey of Methods and Systems ”, Journal of Computer Science 3 (7): 478-486.

49. Shyara Taruna R., (2013),“Enhanced Naïve Bayes Algorithm for Intrusion Detection in Data Mining” , (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (6) , 2013, 960-962

50. Snort user manual. http://www.snort.org/docs, Accessed date Jan 2012

51. Sourcefire |network security solutions www.sourcefire.com

52. S.S. Joshi (2010). “A Study of Information Security Policies in Selected IT Companies in Pune City”. Published Doctorial dissertation, University of Pune. Retrieved March 3, 2011 from http://shodhganga.inflibnet.ac.in/handle /10603/2026.

53. Strata guard, www.securitywizardry.com Accessed date Jan 2012

54. Sunita Beniwal , Jitender Arora,(2012), “Classification and Feature Selection Techniques in Data Mining” , International Journal of Engineering Research & Technology (IJERT),2012

55. T. Lappas and K. P., (2007), “Data Mining Techniques for (Network) Intrusion Detection System”, 2007.

56. Thair Nu Phyu,(2009), “Survey of Classification Techniques in Data Mining” ,Proceedings of the International MultiConference of Engineers and Computer Scientists Hong Kong ,2009 .

57. Tim lane,(2007), “ Information security management in Australian universities :an exploratory analysis” ,thesis, qft faculty of information technology.

58. U. Fayyad, D. Haussler, and P. Stolorz.(1996), “From Data Mining to Knowledge Discovery in Databases” , 0738-4602-1996.

59. Wenke Lee,(1996), “A Data Mining Framework for Building Intrusion Detection Models”,DAPRA.

60. Wenke Lee, (2002),“Applying Data Mining to Intrusion Detection: the Quest for Automation, Efficiency, and Credibility”, SIGKDD.

61. Witten IH, Frank E. , (2005),“ Data Mining: Practical Machine Learning Tools and Techniques” , Second edition, Morgan Kaufmann,2005.

Chapter 3

Research Design and Methodology

3.1. Introduction

This research aims to study network security issues through survey method. Survey conducted for Pune IT companies is intended to study challenges to intrusion detection for computer network security. Survey is conducted by questionnaire method. This research investigates applicability of data mining techniques for intrusion detection. To investigate this, experiment method is used. Various experiments are performed using machine learning software to know efficient methods for intrusion detection.

3.2. Statement of the research problem

Computer network security is the necessity of all IT companies with growing network. For network security one of the most critical factors is detection of intrusion attack on computer security. Intrusion detection is becoming a challenging task due to increased connectivity of computer system and services.

In this context “What are challenges to intrusion detection for computer network security?” is the question to be tackled. Researcher seeks to study network security issues, specifically need of intrusion detection systems and challenges to intrusion detection system to ensure computer network security in IT industrial units of Pune region.

This study is further intended to investigate how data mining techniques can serve for strengthening security. There is need to study how data mining can provide a mechanism to detect intrusion. What data mining techniques are useful to handle

challenges of intrusion detection? For this various experiments using data mining methods are required to execute. These experiments are aiming to find out methods to resolve network security issue effectively. Aim of this study is to provide a framework which is capable to give solution for challenges to intrusion detection.

This research intends to get answers for the following research questions.

1. What are the challenges to current intrusion detection systems?
2. What are the effective data mining techniques for intrusion detection?
3. Why computer network security is essential?
4. How to distinguish whether incoming network traffic is normal or intrusion.
5. How intrusion detection plays important role in computer network security?

3.3. Rational of the study

IT industrial units need to manage security of computer network. Network security is an important factor of IT industrial units. Computer and computer network security becomes integral parts of all IT industries because of increased requirement of network and processing speed.

As the network dramatically extended, security is considered as a major issue in computer networks. Internet attacks are increasing, and there have been various attack methods, consequently. The rapid development of Pune IT industries and growing network facilities makes computer security a critical issue. Because IT industrial units keep important and classified information on their computers, there is a great need to protect that information from those who would exploit it. One way to identify attack is by using IDS, which are designed to locate and alert systems administrators about the presence of malicious traffic.

This study suggests how computer network security management can get benefit of data mining techniques for intrusion based security attack detection. The outcome of this study will also add to the body of knowledge on computer network security

management. The output of this study may also be used as a complementary approach to signature based intrusion detection methods.

3.4. Objective of study

General objective

The general objective of this study is constructing a data mining framework for intrusion detection system that will enhance the network security system.

Specific objectives

1. To study and examine
    - Network security importance and issues in IT industrial units of Pune region.
    - Importance of intrusion detection system and challenges to current intrusion detection systems for network security management.
2. To analyze, computer network security components. Specifically intrusion attack and intrusion detection system.
3. To analyze, several steps involved in data mining process.
4. To analyze, the applicability of existing data mining techniques.
5. To propose data mining techniques through creation of data analysis framework.

3.5. Hypothesis of study

The study is also undertaken to test following hypothesis-

1. Intrusion based security attack has become global challenge to IT sector.
2. Intrusion detection systems are essential for computer network security.

3. Accurate detection of intrusion attack carries immense value in security management and Current ids needs improvement in accuracy of intrusion detection

3.6. Research methodology

This research employs survey method to identify network security issues and experiment method for construction of framework. This research study is related to Network Security Management - A study with special reference to IT industrial units in Pune region. In this study primary and secondary data is collected to find out importance of network security and intrusion detection system. Primary data is collected through survey method whereas secondary data is collected through published and unpublished material. Research methodology [2] used in this research explains process of obtaining sample and size of sample.

3.6.1 Primary data

This data is collected through survey method. This data is original in nature. This data is collected by distributing the questionnaire & getting filled by the concerned respondents, for this purpose, online questionnaire as well as manual method was used. Telephonic and/or personal interview conducted with the IT industry people of Pune region.

3.6.2 Sample Design

Sample design is a specific plan which designed to get samples from population. To serve the purpose of the research subject, the researcher has selected the total 30 sample units. Sampling technique used is Purposive Quota and convenience sampling. Population for this study is IT industrial units from Pune region. Size of population is 200 IT industries. Sampling frame used for the study is 30 IT industrial units. Parameter of interest for this study: Determining need and challenges of computer network in selected IT industrial units.

3.6.3 Selection of region

The researcher has used purposive sampling method to select region for the study. Researcher has selected Pune region because it is attractive destination for IT industrial units. IT industries are growing in Pune because of its close proximity to Mumbai and rapidly growing infrastructure. Large number of educational institutes and universities also give a reason for growing IT industries. This region is IT hub and most of the leading IT companies have branch in Pune. Along with leading companies many emerging companies are located in Pune[1][5].

3.6.4 Selection of respondent

Respondent are selected those who are working in IT industry with more than two years of experience. Researcher has selected only those employees who actually work on network security or network security related projects. Questionnaire is filled by all the selected employees. This questionnaire is either filled manually or sent through Email. Email questionnaire are manually filled, scanned or filled in softcopy. With this all the employees are interviewed by personal and/or telephonic method.

3.6.5 Secondary data

Secondary data is used to study the network security offered by various products of intrusion detection available in market. It is used to find features and limitations of current IDS products. Secondary data is collected from reputed journals, articles, websites and product documentation.

3.6.6 Questionnaire

Questionnaire is meant for obtaining information about importance and necessity of computer network security measures. Questionnaire specifically designed for network security therefore it further gather information about need of intrusion detection system and investigates challenges to current intrusion detection systems. Questionnaire is created with likert scale [3][4] and multiple choice.

3.6.7 Testing of hypothesis.

Hypothesis testing is a procedure to either accept or reject hypothesis. It recognizes and identifies the relevant facts and gives direction to research study. In this study hypothesis has been tested using percentage.

3.7. Limitations of study

Pune region is IT hub and the researcher being from Pune region; the study is limited to this region only. Conclusions drawn from the survey is limited for IT industry of Pune region only.

- This research specifically deals with only intrusion detection component of computer security.
- This research is mainly focused on computer security management. This aims to provide a mechanism to detect security attacks. This thesis does not offer mechanism to prevent the attack.
- The framework constructed in this thesis just notify for the administrators after detecting an attack and administrators can take action for security management. Being informed properly is the basis of every management, so this thesis informs about detection of security attack.

3.8. Chapter References

1. IT companies in and around Pune , www.punediary.com
2. Kothari C. R., (2004), “Research Methodology, Methods and techniques” (2nd ed.), New Delhi: New age International (p) Ltd.
3. Harry N Boone, Deborah A Boone, (2012),”analyzing likert data”, journal of extension, vol 50.
4. Geoff Norman, (2010), “Likert scales, levels of measurement and the laws of statistics”, Springer Science Business Media B.V.
5. Pune IT software companies in Pune list, www.pune.ws

Chapter 4

Data Analysis and interpretation

4.1. Introduction

This research is related to Network Security Management - A study with special reference to IT industrial units in Pune region. The researcher has tested the hypothesis with the help of primary and secondary data. Primary data is collected through the questionnaire. Statistical and graphical methods are used data analysis. An analysis is carried out under following broad headings

1. Importance of security
2. Why security measures are important?
3. Importance of intrusion detection systems for network security.
4. What are the challenges to current intrusion detection systems?

This data is collected through survey method [2]. This data is original in nature. This data is collected by distributing the questionnaire & getting filled by the concerned respondents, for this purpose, online questionnaire as well as manual method was used. Telephonic and/or personal interview conducted with the IT industry people of Pune region.

The following steps were used for collecting the primary data-

1. Questionnaire is filled by all the selected employees. This questionnaire is either filled manually or sent through Email. Email questionnaire are manually filled, scanned or filled in softcopy. With this all the employees are interviewed by personal and/or telephonic method.
2. Telephonic and personal interviews conducted.

Questionnaire used for study is meant for obtaining information about importance and necessity of computer network security measures. Questionnaire specifically designed for network security therefore it further gather information about need of intrusion detection system and investigates challenges to current intrusion detection systems. Questionnaire used for survey consist questions based on scale. Likert scale provides a statement, which respondent is asked to evaluate. The likert scale used is balanced on both the side of neutral option. Likert scale [2] [3] is used because one of standard scale to collect opinion experiences or specific data.

- **Pune IT industry**

The researcher has used purposive sampling method to select region for the study. Researcher has selected Pune region because region is IT hub and most of the leading IT companies have branch in Pune. Along with leading companies many emerging companies are located in Pune. Sampling technique used is Purposive Quota Sampling. Population for this study is IT industrial units from Pune region. Size of population is 200 IT industries. Sampling frame used for the study is 30 IT industrial units. Parameter of interest for this study: Determining need and challenges of computer network in selected IT industrial units.

- **Background of respondent**

Respondent are selected those who are working in IT industry with more than two years of experience. Researcher has selected only those employees who actually work on network security or network security related projects.

4.2. Network security issues

An attempt was made to meet one of the objectives of this study which is “to study Network security importance and issues in IT industrial units of Pune”. The primary data collected from the respondents from IT industrial units of Pune region. To study importance of security parameters like security threats, what level of confidential data is stored on machine connected through network and relationship between security and cost is surveyed.

4.2.1. Intrusion based security attack

The rise of computer network and emerging technologies made computer network security work very challenging. In spite of various measures of network security, still computer connected through network have high possibility of security attacks. Intrusion based security attacks are viable on machine connected through network. Connection through network is need of hour. To keep a secure network connection is very challenging. Computer network security is very essential. IT industrial units are surveyed whether computer connected through network have possibility of intrusion based security attack or not.

Table 4.1 viability of Intrusion based Security attacks

Possibility of security attack on any computer connected through network	**SA**	**A**	**N**	**DA**	**SD**	**TOTAL**
No. of response	11	15	2	1	1	30
Percentage of response	37%	50%	7%	3%	3%	100
Source : Primary data						

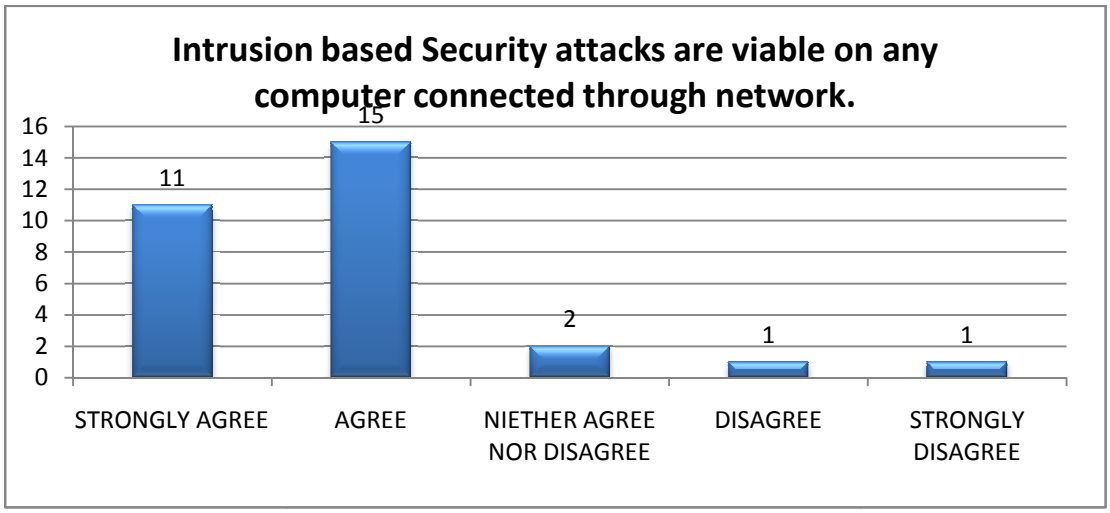
Note: SA- Strongly Agree, A -Agree, N –Neutral (neither agree nor disagree) DA- Disagree, SD- strongly Disagree

The objective of table 4.1 is to know the possibility of security attack on any computer connected through network. It is measured on five point likert scale having items like strongly Disagree, Disagree, Neutral (neither agree nor disagree) Agree, and Strongly Agree . Of the total 30 companies, 87% companies agree or strongly agree that there is strong possibility of security attack to computer, only 7 % neither agree nor disagree and 6% disagree or strongly disagree.

The chart 4.1 indicates the possibility of security attack on any computer connected through network. Of the total 30 companies, 26 companies agree or strongly agree

that there is strong possibility of intrusion based security attack to computer, only 2 neither agree nor disagree and 2 disagree or strongly disagree.

Chart 4.1: Response to likert scale used to know about possibility of Intrusion based Security attacks



4.2.2. Why network security is important?

Importance of computer network security is extremely important one of the major reason for this is confidential data is stored on the computers. Definitely more you keep valuables at your house more you are concerned about house security. Similarly if highly confidential data is stored on computer than security is more indispensable. To understand this survey is done to inquire, do confidential data is stored on computers of IT industrial units**?**

The table 4.2 presents that confidential data is stored on the computers of organization. Of the total 30 companies, 59% respondents agree or strongly agree that highly confidential data is stored in their computer, only 17% neither agree nor disagree and 24 % disagree or strongly disagree.

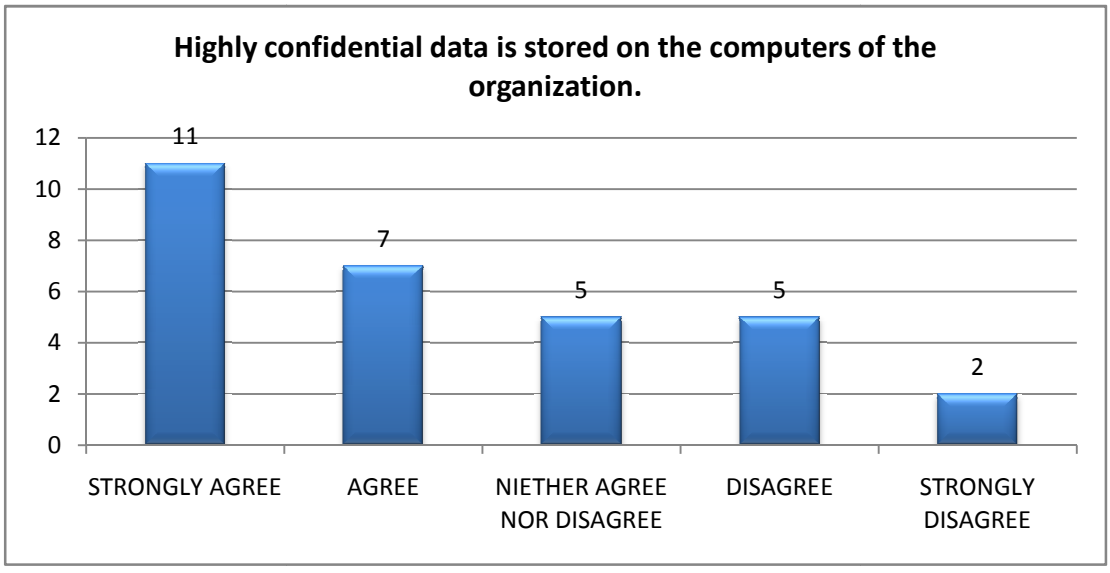
Table 4.2 Highly confidential data is stored on the computers of the organization.

Confidential data is stored on computers	SA	A	N	DA	SD
No. of response	11	7	5	5	2
Percentage of response	36%	23%	17%	17%	7%
Source : Primary data					

Note: SA- Strongly Agree, A -Agree, N –Neutral (neither agree nor disagree) DA- Disagree, SD- strongly Disagree

The chart 4.2 presents that confidential data is stored on the computers of organization. Of the total 30 respondents, 18 respondents agree or strongly agree that highly confidential data is stored in their computers of their organization, only 5 neither agree nor disagree and 7 disagree or strongly disagree.

Chart 4.2: Response to likert scale used to about confidential data is stored on computers



4.2.3. Does compromise with security affects cost?

Other than data confidentiality one most important reason for requirement of security is cost and financial factors. Compromise with security affects cost. Compromise with security increases cost like hardware cost, software cost, maintenance cost, cost of data loss and cost of incorrect decision making.

It can be observed through the table 4.3 that security is associated with cost. Of the total 30 companies, 100% () companies agree or strongly agree that Computer network security is very essential because Compromise with security affects cost.

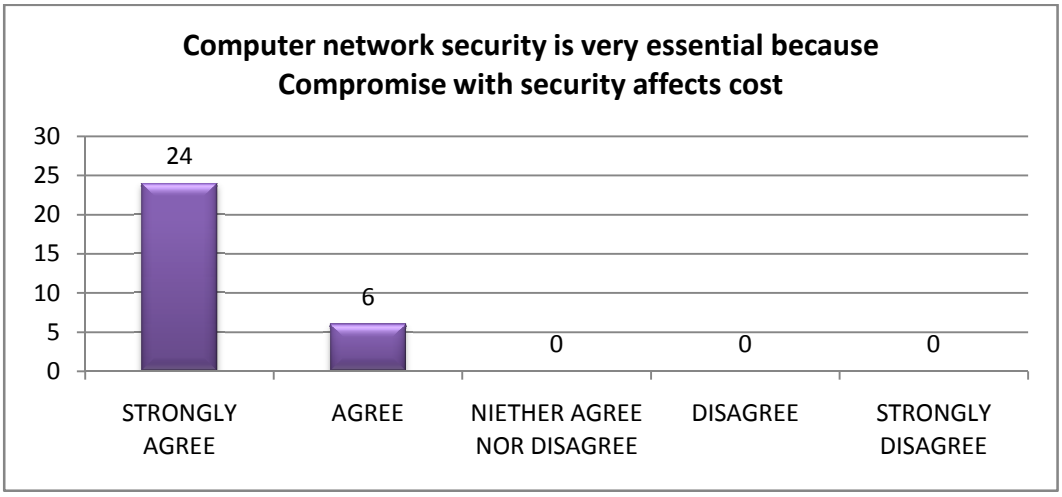
Table 4.3 **Negligence in security affect cost**

Negligence in security affect cost	**SA**	**A**	**N**	**DA**	**SD**
No. of response	24	6	0	0	0
Percentage of response	80%	20%	0%	0%	0%
Source : Primary data					

Note: SA- Strongly Agree, A -Agree, N –Neutral (neither agree nor disagree) DA- Disagree, SD- strongly Disagree

Chart 4.3 represents that security is associated with cost. Of the total 30 respondents, all 30 respondents agree or strongly agree that Computer network security is very essential because Compromise with security affects cost.

Chart 4.3: Response to likert scale used to know relationship between computer security and cost



Accountability of security

Usually it is assumed that computer network security is accountability of network admin or security employees but from the table interesting observation can be made.

Table 4.4 Accountability of Computer security in the organization.

Security is accountability of everyone in organization	**SA**	**A**	**N**	**DA**	**SD**
No. of response	21	8	1	0	0
Percentage of response	70%	27%	3%	0%	0%
Source : Primary data					

Note: SA- Strongly Agree, A -Agree, N –Neutral (neither agree nor disagree) DA- Disagree, SD- strongly Disagree

The objective of this table 4.4 is to know what respondent think and experience about accountability about computer network security. 97% companies agree or strongly agree that network security is accountability of everyone in the organization ,only 3% neither agree nor disagree and no one disagree or strongly disagree .

Chart 4.4 shows Of the total 30 companies, 27 companies agree or strongly agree that network security is accountability of everyone in the organization

Chart 4.4: Response to likert scale used to know about accountability of computer security.

4.3. Importance of intrusion detection system

Generally it is considered that if we have antivirus our computers are secure but if we have firewall along with antivirus our computer network is completely secure. To understand this, a five point likert scale is used having items like scale having items like strongly Disagree, Disagree, Neutral (neither agree nor disagree) Agree, and Strongly Agree.

Very interesting observation is done through this survey that only 23% agree or strongly agree that having antivirus along with firewall is sufficient to makes computer network completely secure. Whereas 64% of respondent disagree or strongly disagree.

The table 4.5 shows whether popular security software is sufficient to secure computer completely. Of the total 30 companies, companies 23% agree or strongly agree that Having both antivirus and firewall is sufficient to makes your computer network completely secure, only 13% niether agree nor disagree and 64% disagree or strongly disagree .

Table 4.5 Security components to make computer network completely secure.

Having both antivirus and firewall makes your computer network completely secure.	**SA**	**A**	**N**	**DA**	**SD**
No. of response	0	7	4	15	4
Percentage of response	0%	23%	13%	50%	14%
Source : Primary data					

Note: SA- Strongly Agree, A -Agree, N –Neutral (neither agree nor disagree) DA- Disagree, SD- strongly Disagree

Chart 4.5.: Response to likert scale used to know that use of antivirus and firewall is sufficient for complete security

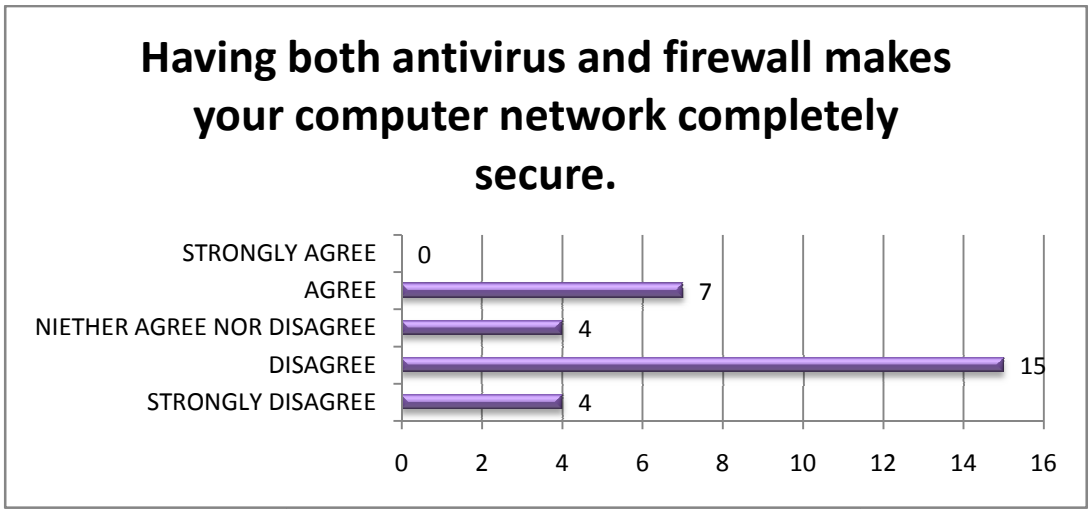


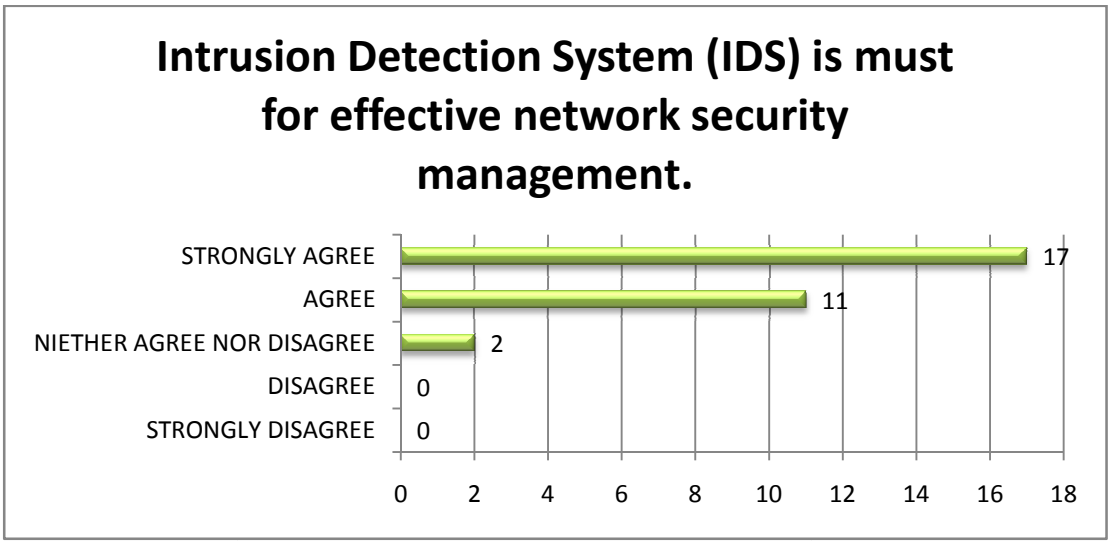
Table 4.6 Importance of Intrusion Detection System (IDS)

IDS is must for network security	**SA**	**A**	**N**	**DA**	**SD**
No. of response	17	11	2	0	0
Percentage of response	57%	36%	7%	0%	0%
Source : Primary data					

Note: SA- strongly Agree, A- Agree, N –Neutral (neither agree nor disagree) D -Disagree, SD- Strongly Disagree

The table 4.6 gives information about importance of intrusion detection system for security management. Of the total 40 companies, 93% companies agree or strongly agree that IDS intrusion detection system is must for computer network security, only 7% neither agree nor disagree and % disagree or strongly disagree .

Chart 4.6: Response to likert scale used to know how essential IDS are



Anomaly Based IDS versus Signature Based IDS

Two popular categories of intrusion detection systems are available ;Anomaly Based IDS and Signature Based IDS(SB- IDS). intrusion detection products perform signature analysis. Signature analysis is pattern matching of system settings and user activities against a database of known attacks. Anaomaly based IDS perform analysis finds variation from normal patterns of network behavior. Possible intrusions are signalled when observed values fall outside the normal range.

The table 4.7 depict that which type of intrusion detection system more useful for the companies. Of the total 30 companies, 80% companies ,agree or strongly agree that Anomaly Based IDS (AB-IDS) are more suitable for our organization than Signature Based IDS(SB- IDS), only 20 % neither agree nor disagree and no one disagree or strongly disagree .

Table 4.7 Anomaly Based IDS versus Signature Based IDS.

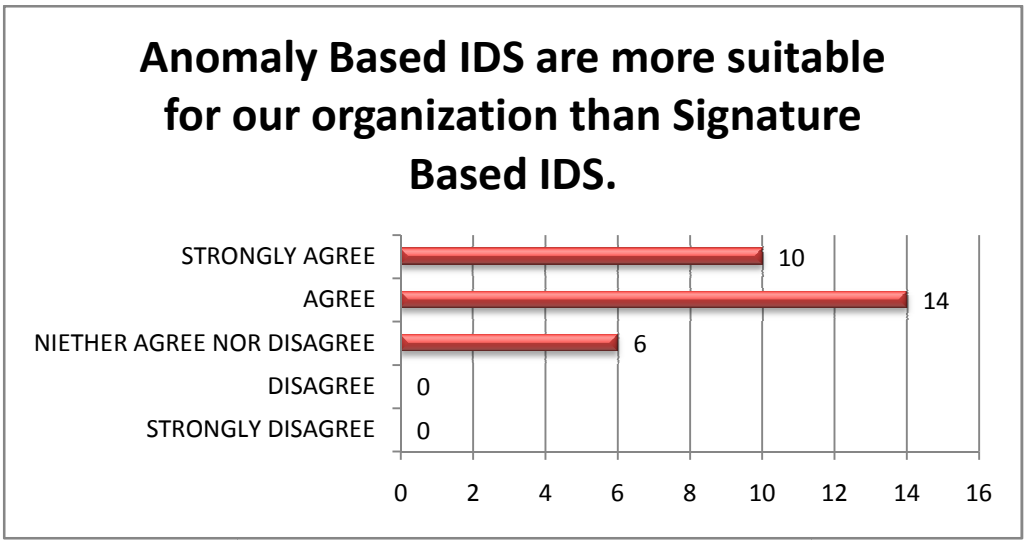
Anomaly Based IDS are better than Signature Based IDS	**SA**	**A**	**N**	**DA**	**SD**
No. of response	10	14	6	0	0
Percentage of response	33%	47%	20%	0%	0%
Source : Primary data					

Note: SA- strongly Agree, A- Agree, N –Neutral (neither agree nor disagree) D -Disagree, SD- Strongly Disagree

Chart 4.7 depict that which type of intrusion detection system more useful for the companies. Of the total 30 companies, 24 companies ,agree or strongly agree that Anomaly Based IDS (AB-IDS) are more suitable for our organization than Signature

Based IDS(SB- IDS),only 20 % neither agree nor disagree and no one disagree or strongly disagree .

Chart 4.7: Response to likert scale used to know Anomaly based IDS are better than signature based IDS.



4.4. Issues related to intrusion detection system

This research intends to study implementation and monitoring issues of Intrusion detection system. Issues identified are threats to computer network security , Challenge to intrusion detection system and Important parameter to for selection of intrusion detection system.

- **Threats to computer network security**

There are various threats to computer network security like Virus/worm attack, unauthorized access, malicious attack, Denial of service. This study made an attempt to know which one is most critical computer network security threat.

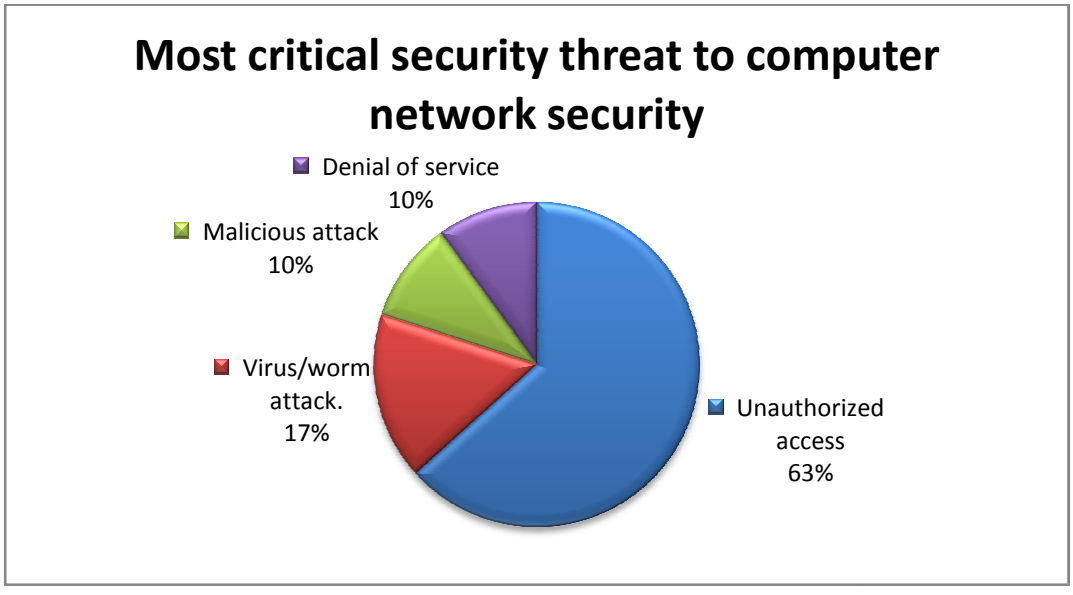
The table illustrate that out of 30 respondents ,19 respondents identifies most critical security threat is Unauthorized access, 5 respondents considers that Virus/worm attack are critical, 3 respondents Malicious attack and , remaining 3 respondents considers Denial of service attack very critical.

Table 4.8 Most critical security threat to computer network security

Sr. No.	Most critical security threat to computer network security?	No of respondents	Response in percentage
1	Unauthorized access.	19	63%
2	Virus/worm attack.	5	17%
3	Malicious attack	3	10%
4	Denial of service	3	10%
Total		30	100%
Source : Primary data			

The chart 4.8 illustrate that 63% respondents identifies most critical security threat is Unauthorized access, 17 % believe Virus/worm attack, 10 % Malicious attack and , remaining 10 % Denial of service.

Chart 4.8: Most critical threat to network security



- **Most critical challenge to intrusion detection system**

Intrusion detection system provides next layer to security, but there are many challenges. Identifying type of intrusion (IDS must rightly identify intrusion type), false alarm about attack (false alarm means either attack is detected normal or normal data is identified as attack), alerting mechanism (user friendly), updating signature policy (signature database must be regularly updated) etc are

The table 4.9 illustrate that most critical challenge for intrusion detection system as per 16 respondents is false alarm . 10 respondents consider identifying type of intrusion, 3 respondents consider alerting mechanism whereas only 1 respondent says updating signature policy.

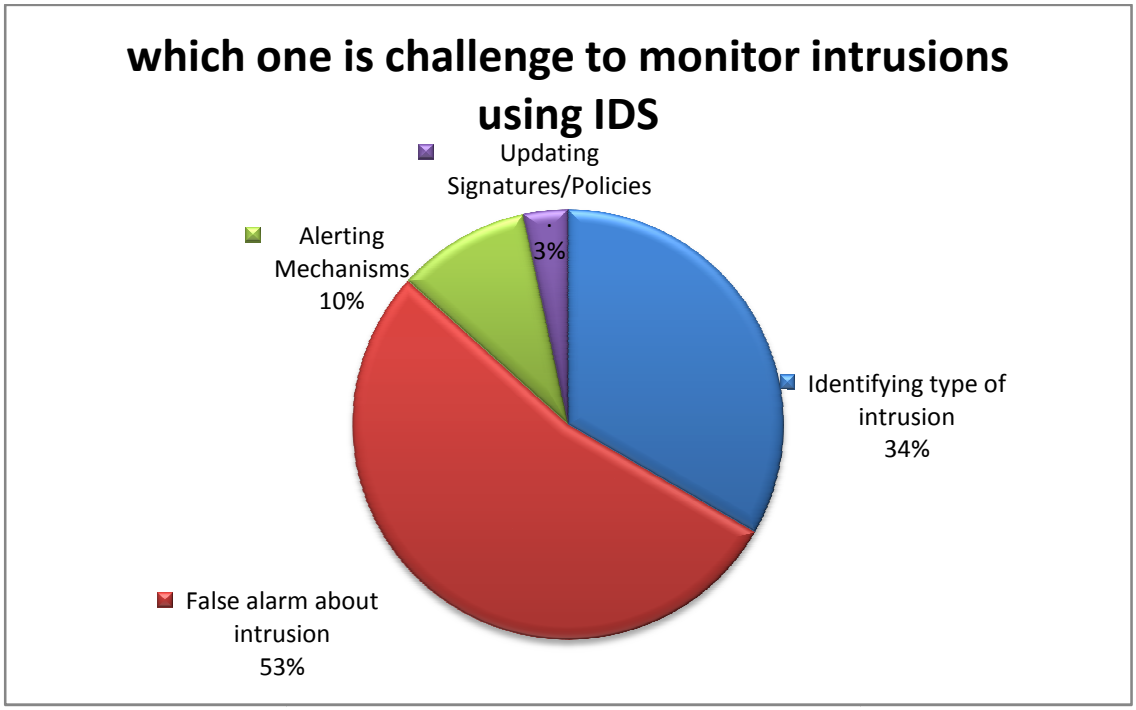
Table 4.9 Most critical challenge to monitor intrusions using IDS

Sr. No.	Most critical challenge to monitor intrusions using IDS?	No of respondents	Response in percentage
1	Identifying type of intrusion	10	34%
2	False alarm about intrusion.	16	53%
3	Alerting Mechanisms.	3	10%
4	Updating Signatures/Policies.	1	3%
Total		30	100%
Source : Primary data			

The table illustrates that most critical challenge for intrusion detection system as per 53% Pune IT industrial units is false alarm about intrusion. 34% respondents consider

identifying type of intrusion, 10% respondents consider alerting mechanism whereas only 3% respondent says updating signature policy.

Chart 4.9 Most critical challenge to monitor intrusions using IDS



- **Important parameter to for selection of intrusion detection system.**

There are many parameters which are considered for selection of intrusion detection system. Parameters like how popular the IDS product is, whether it has capacity to detect new intrusion, easy and user friendly user interface, what is accuracy of intrusion detection are considered important.

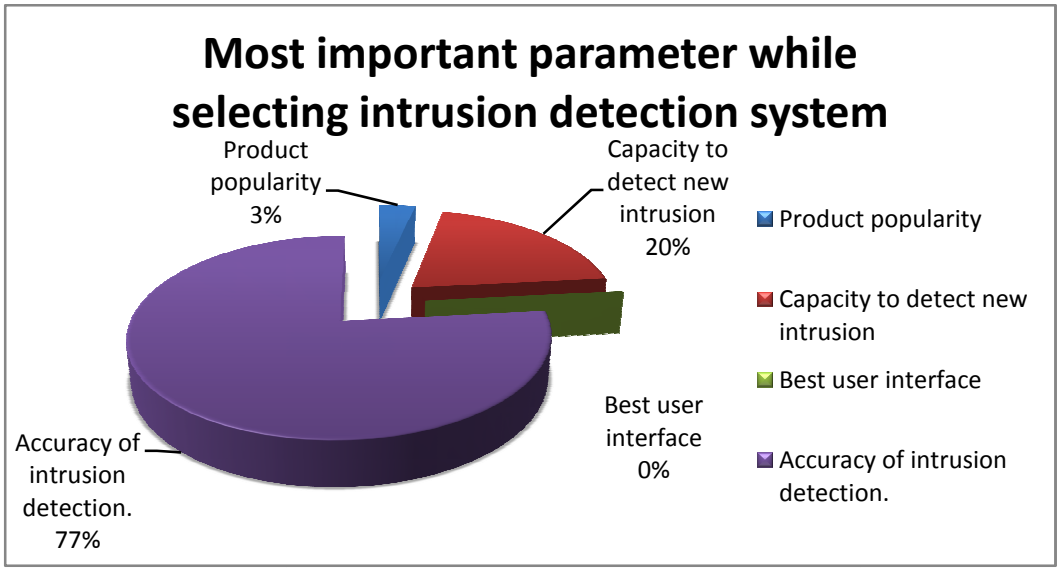
The table 4.10 demonstrate that most important parameter is accuracy of intrusion detection . of the 30 respondents , 23 respondents says Accuracy of intrusion detection is most important . 1 respondent consider product popularity very important whereas 6 respondent think capacity to detect new intrusion is imperative. No one consider best user interface important parameter for selection of IDS.

Table 4.10 Most important parameter while selecting intrusion detection system

Sr. No.	Most important parameter while selecting intrusion detection system for the security management of your organization?	No of respondents	Response in percentage
1	Product popularity	1	3%
2	Caparegion to detect new intrusion	6	20%
3	Best user interface	0	0%
4	Accuracy of intrusion detection.	23	77%
		30	100%
Source : Primary data			

The chart 4.10 shows respondents 77% says Accuracy of intrusion detection is most important . 3% respondents consider product popularity very important whereas 20% respondent think capacity to detect new intrusion is imperative.

Chart 4.10: Most important parameter for selection of IDS



4.5. Testing of hypothesis

Hypothesis 1.

The first hypothesis of the study is **"Intrusion based security attack has become global challenge to IT sector".**

This hypothesis has been tested by using percentage. To study this, parameter like possibility of security attack on computer connected through network, confidential data, cost security relationship, security accountability is considered. To understand in depth, study is done, to recognize which security attack is most crucial for IT industrial units of Pune Region.

Table 4.11: Network security issues survey

Network security issues	SA	A	N	D	SD	Total
Intrusion based security attacks are viable on computer	11	15	2	1	1	30
Highly Confidential data is stored on computers	11	7	5	5	2	30
Negligence in security affects cost	24	6	0	0	0	30
Security is accountability of everyone in the organization	21	8	1	0	0	30
Total (percentage)	67	36	8	6	3	120
Percentage	55.83%	30.00%	06.66%	05.00%	02.50%	100%
Source : Primary Data (30 IT industrial units of Pune region)						

Note: SA- strongly Agree, A- Agree, N –Neutral (neither agree nor disagree) D - Disagree, SD- Strongly Disagree

85.83% Respondent agree or strongly agree that network security is essential, it is a global challenge to IT industrial units and 6.6 are Neutral whereas 7.5 disagree of strongly disagree.

It is observed that majorities of companies considers Intrusion based security attack has become global challenge to IT sector. 85.83 % agree or strongly agree for this.

Therefore it is concluded that 'Intrusion based security attack has become global challenge to IT sector'. **Hence hypothesis of the study is accepted.**

Hypothesis 2

Second hypothesis of the study is **"Intrusion detection systems are essential for computer network security"**

This hypothesis has been tested by using percentage. 64% companies disagree or strongly disagree that having both antivirus and firewall is sufficient to makes your computer network completely secure. It means companies do not rely only on antivirus, firewall for maintaining secure network. They use other security components also. Further 93% companies agree or strongly agree that IDS intrusion detection system is must for computer network security.

Thus, Intrusion detection systems are highly required for effective computer network security. Therefore we accept the Hypothesis and conclude that Intrusion detection systems are essential for computer network security. **Hence hypothesis of the study is accepted.**

Hypothesis 3.

Third hypothesis of the study is **"Accurate detection of intrusion attack carries immense value in security management and Current IDS needs improvement in accuracy of intrusion detection"**

To study this, study is done on the basis of most important parameter for selection of intrusion detection system is considered. Along with this study is done to identify most critical challenge for intrusion detection system is taken.

This hypothesis has been tested by using percentage. 77% companies say 'Accuracy of intrusion detection' is most important for selection of IDS.

53% Pune IT industrial unit identifies false alarm generation as most critical challenge for intrusion detection system. False alarm are directly associated with accuracy of IDS. If accuracy of IDS is high means less false alarms are generated.

Therefore we accept the Hypothesis and conclude that ‘Accurate detection of intrusion attack carries immense value in security management, Current ids needs improvement in accuracy of intrusion detection’

4.6. Chapter references

1. Kothari C. R. ,(2004), “Research Methodology, Methods and techniques” (2nd ed.), New Delhi: New age International (p) Ltd.

2. Harry N Boone , Deborah A Boone, (2012),”analyzing likert data”, journal of extension, vol 50.

3. Geoff Norman, (2010), “ Likert scales, levels of measurement and the laws of statistics”, Springer Science Business Media B.V.

Chapter 5

Experiments execution and Design of Framework

5.1. Introduction

This is an experiment based research, this chapter elaborates experimental design, details of experiment performed on network data. Previous chapter demonstrate that accuracy is vital for intrusion detection. To provide accurate intrusion detection ,new framework is required .This chapter elaborates experimentation performed for intrusion detection followed by analysis. Further, results are analyzed on the basis of performance measurement terms like correctly classified instances, true positive rate, false positive rate etc. and finally give details about designs of framework for detection of intrusion attack to strengthen computer network security

5.2. Experiment design

The specific questions that the experiment is intended to answer must be clearly identified before carrying out the experiment. In this research work analysis is done on data collected from experiment. It is wise to take time and effort to organize the experiment properly to ensure that the right type of data, and enough of it, is available to answer the questions of interest as clearly and efficiently as possible. This process is called experimental design .

Primary objective of this study is to build framework for intrusion detection using data mining. Experiments are performed using following design strategy.

1. Selection of dataset.
2. Recognize types of intrusion attack and their specification.
3. Applying appropriate data preprocessing techniques.
4. Applying appropriate classification technique.
5. Performance measurement terms for evaluation of classifier performance.
6. Choosing best classifier to build model for intrusion detection.

In this experimental research initially data set is taken for experiments. Data set have different Features available and types of attack like DoS, Probe and U2R. Data preprocessing techniques like data cleaning, replacing missing value, removing redundant and unnecessary attributes is applied on dataset. Feature selection techniques are used to select most relevant features for intrusion detection. For experimentation, classification methods like decision tree, Bayes net, rule based classifier, ensemble methods are planned to use. Decision tree based J48 algorithm, rule based OneR algorithm and bayes based bayes net algorithms are taken as classifiers. Classifier performance is evaluated based on performance measurement metrics like correctly classified instance, relative error, absolute error, True positive rate, false positive rate and confusion matrix . weka software is used for experimentation

In this study, the experiments were conducted following the Knowledge Discovery in Database process model. The Knowledge Discovery in Database process model starts from selection of the datasets. The dataset used in this study has been taken from KDD Cup dataset available on line. The major preprocessing activities include fill in missed values, remove outliers; resolve inconsistencies, integration of data that contains both labeled and unlabeled datasets, feature reduction . Attribute selection methods are applied .A total of 21,533 intrusion records are used for training the models. For validating the performance of the selected model, a separate 3,397 records are used as a testing set.

10 experiments are planned to performed using following classifiers, validation methods, preprocessing and ensemble methods

- For experimentation 3 basic classifiers are used.
  - ❖ Decision tree classifier
  - ❖ Rule based classifier
  - ❖ Bayes net classifier
- All the three types of classifiers are tested with two validation strategies
  - ❖ Percentage split
  - ❖ 10 fold cross validation.
- Two data preprocessing filters are used.
  - ❖ Supervised attribute selection
  - ❖ Discritization
- Two ensemble methods are used
  - ❖ Bagging
  - ❖ Boosting

5.2.1. Performance measurement terms

To evaluate performance of classifier, performance measurement terms are following

1) *Correctly classified instance*

The correctly and incorrectly classified instances show the percentage of test instances that were correctly and incorrectly classified. The percentage of correctly classified instances is called accuracy or sample accuracy.

2) *Kappa statistics*

Kappa is a chance-corrected measure of agreement between the classifications and the true classes. Kappa statistics is calculated by taking the agreement expected by chance away from the observed agreement and dividing by the maximum possible agreement. A value greater than 0 means that classifier is doing better than chance.

3) *Mean absolute error, Root mean squared error, Relative_absolute_error*

The error rates are used for numeric prediction rather than classification. In numeric prediction, predictions aren't just right or wrong, the error has a magnitude, and these measures reflect that.

Confusion matrix

A confusion matrix contains information about actual and predicted classifications done by a classification system. Performance of such systems is commonly evaluated using the data in the matrix. The following table shows the confusion matrix for a two class classifier. The entries in the confusion matrix have the following meaning in the context of this study:

- *a* is the number of **correct** predictions that an instance is **negative**,
- *b* is the number of **incorrect** predictions that an instance is **positive**,
- *c* is the number of **incorrect** predictions that an instance **negative**, and
- *d* is the number of **correct** predictions that an instance is **positive**.

Table 5.1 confusion matrix

		Predicted	
		Negative	**Positive**
Actual	**Negative**	A	B
	Positive	C	D

Several standard terms have been defined for the 2 class matrix:

- The *accuracy* (*AC*) is the proportion of the total number of predictions that were correct. It is determined using the equation:

$$AC = \frac{a+d}{a+b+c+d}$$

- The *recall* or *true positive rat*e (*TP*) is the proportion of positive cases that were correctly identified, as calculated using the equation:

$$TP = \frac{d}{c+d}$$

- The *false positive rate* (*FP*) is the proportion of negative cases that were incorrectly classified as positive, as calculated using the equation:

$$FP = \frac{b}{a+b}$$

- The *true negative rate* (*TN*) is defined as the proportion of negative cases that were classified correctly, as calculated using the equation:

$$TN = \frac{a}{a+b}$$

- The *false negative rate* (*FN*) is the proportion of positives cases that were incorrectly classified as negative, as calculated using the equation:

$$FN = \frac{c}{c+d}$$

- Finally, *precision* (*P*) is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$P = \frac{d}{b+d}$$

5.2.2. Data set-NSL KDD

The data set used to perform the experiment in this research is taken from NSL-KDD Data set. The data set was chosen to assess rules and to detect intrusion. This dataset [2] contains 41 features.

NSL-KDD is improved version KDD cup 99 . KDD Cup which is widely accepted as a standard dataset for research work. NSL-KDD dataset is public dataset available for research work ,it represents picture of real world network data. In addition, the NSL-KDD dataset has rational number of records in train and test sets. This offers a reasonable way to run the experiments. Consequently, assessment results of different research work will be consistent and comparable.

In NSL KDD dataset all the connections are labelled as normal or attacks. Attacks falls into 4 major categories.

1. DOS :- Denial of Service
2. Probe :- Gather information about the targeted network
3. U2R :- unauthorized access to root privileges,
4. R2L :- unauthorized remote login to machine.

In this dataset, features can be categorized in 3 groups namely Basic features, content based features and time based features.

- This dataset have two types of sets namely training set and test set. Training set has approx 50, 00,000 connections whereas Test set has 3, 00,000 connections. There are many attack types, which are provided in Test data are not available in the training data. This gives more realistic picture of real world. Train set have 22 attack types. Test data have additional 17 new attack types that belong to one of four major categories.

Features available in KDD dataset

These features can be categorized as follows.

Time based features have two types same host features and same service features .The ‘same host’ features examine protocol behavior ,service etc. it inspect the connections in the past two seconds that have the same destination host as the current connection. The related ‘same service’ features inspect only the connections in the past two seconds that have the same service as the current connection.

Host based traffic features .Some probing attacks scans the host computer by taking larger time interval. therefore, connection records were also sorted by destination host, and features were constructed using a window of 100 connections to the same host instead of a time window.

R2L and U2R attacks do not have sequential patterns that are frequent in most of the DOS and probing attacks. DOS and probe attacks occupy many connections to some host in a very small period of time, whereas R2L and U2R attacks usually occupy only single connection.

Content features like number of failed login attempts are analyzed by an algorithms which uses domain knowledge to add features that look for suspicious behavior in the data portions.

A complete listing of the set of features defined for the connection records is given below.

Basic features of individual TCP connections are:-

- Duration

  This feature represents length (number of seconds) of the connection.

- Protocol_type

  This feature represents type of the protocol, e.g. Tcp, udp, etc.

- Service

  This feature represents network service on the destination, e.g., http, telnet, etc.

- Source_bytes

  This feature represents number of data bytes from source to destination

- Destination_bytes

  This feature represents number of data bytes from destination to source

- Flag

  This feature represents normal or error status of the connection

- Land

  This feature represents whether it is from/to same host/port. Value is 1 if

  Connection is from/to the same host/port otherwise value is 1.

- Wrong_fragment

  This feature represents number of ‘wrong’ fragments

- Urgent

  This feature represents number of urgent packets

Content features within a connection suggested by domain knowledge.

- Hot

  This feature represents number of ‘hot’ indicators

- Num_failed_logins

  This feature represents number of failed login attempts

- Logged_in

  This feature represents value 1 if successfully logged in, otherwise value is 0

- Number_compromised

  This feature represents the number of ‘compromised’ conditions

- Root_shell

  This feature represents 1 if root shell is obtained, otherwise gives 0.

- Su_attempted

  This feature represents 1 if ‘su root’ command attempted; 0 otherwise

- Number_root

  This feature represents the number of root accesses.

- Num_file_creations

  This feature represents number of file creation operations.

- Num_shells

  This feature represents number of shell prompts.

- Num_access_files

  This feature represents number of operations on access control files .

- Num_outbound_cmds

  This feature represents number of outbound commands in an ftp session

- Is_hot_login

  This feature represents 1 if the login belongs to the ‘hot’ list; 0 otherwise

- Is_guest_login

  This feature represents 1 if the login is a ‘guest’ login; 0 otherwise

Traffic features

- Count

  This feature represents number of connections to the same host as the current connection in the past two seconds

The following features refer to same-host connections.

- Serror_rate

This feature represents the percentage of connections that have ‘syn’

Errors

- Rerror_rate

  This feature represents the percentage of connections that have ‘rej’

  Errors

- Same_srv_rate

  This feature represents the percentage of connections to the same

  Service

- Diff_srv_rate

  This feature represents the percentage of connections to different

  Services

- Srv_count

  This feature represents number of connections to the same service as the current connection in the past two seconds.

The following features refer to these same-service connections.

- Srv_serror_rate

  This feature represents the percentage of connections that have ‘syn’

  Errors

- Srv_rerror_rate

  This feature represents the percentage of connections that have ‘rej’ errors

- Srv_diff_host_rate

  This feature represents the percentage of connections to different hosts

The new version of KDD data set, NSL-KDD is publicly available for researchers through website. It can be applied as an effective standard data set to help researchers compare different intrusion detection methods.

There is strong need of useful algorithms for mining the unstructured data automatically.

5.2.3. Brief description of Weka software

WEKA (Waikato Environment for Knowledge Analysis) [3] software offers data mining tasks by a collection of machine learning algorithms. It contains tools for data preprocessing, classification, regression, clustering, association rules, and visualization. For this research purpose, the classification tools were used.

WEKA has four different modes to work in.

• Simple CLI; provides a simple command-line interface that allows direct execution of WEKA commands.

• Explorer; an environment for exploring data with WEKA.

•Experimenter; an environment for performing experiments and conduction of statistical tests between learning schemes.

• Knowledge Flow; presents a 'data-flow' inspired interface to WEKA.

For most of the tests performed for this research work, which will be explained in more detail later, the explorer mode of WEKA is used. Main screen of weka software is shown in figure 5.1.

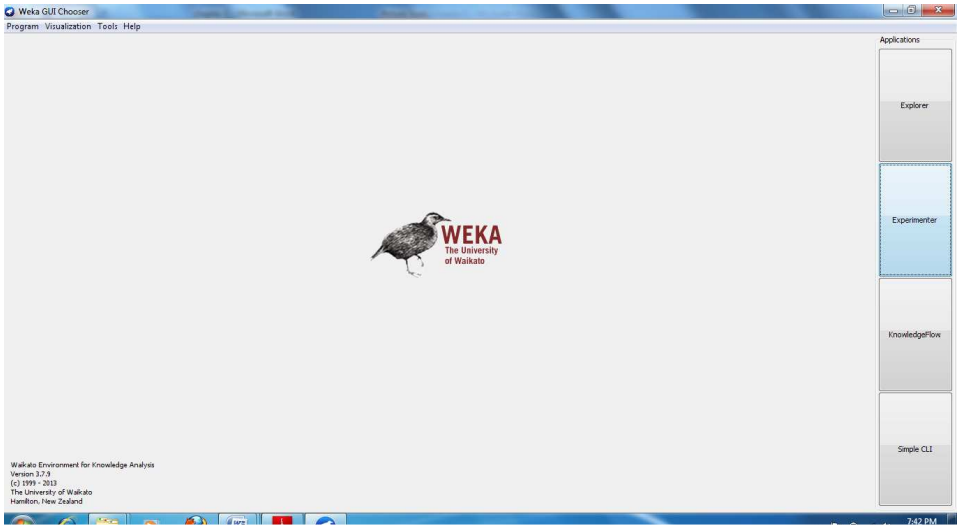


Figure 5.1 Weka software main screen

Weka software [3] needs its input data in ARFF format . ARFF file format [1] is explained below.

- A dataset has to start with a declaration of its name:
  @relation name
- This is followed by a list of all the attributes in the dataset (including the predicted attribute). These declarations have the form:
  @attribute attribute_name specification
- If an attribute is nominal, specification contains a list of the possible attribute values in curly brackets:
  @attribute nominal_attribute {first_value, second_value, third_value}
- If an attribute is numeric, specification is replaced by the keyword numeric: (Integer values are treated as real numbers in WEKA.)
  @attribute numeric_attribute numeric
- In addition to these two types of attributes, there also exists a string attribute type. This attribute provides the possibility to store a comment or ID field for each of the instances in a dataset:
  @attribute string_attribute string
- After the attribute declarations, the actual data is introduced by a tag:
  @data

- This is followed by a list of all the instances. The instances are listed in comma-separated format, with a question mark representing a missing value.
- Comments are lines starting with %

Following is the structure of Arff file used for experiment.

@relation 'NSL-KDD' @attribute 'duration' real

@attribute 'protocol_type' {'tcp','udp', 'icmp'}

@attribute 'service' {'aol', 'auth', 'bgp', 'courier', 'csnet_ns', 'ctf', 'daytime', 'discard', 'domain', 'domain_u', 'echo', 'eco_i', 'ecr_i', 'efs', 'exec', 'finger', 'ftp', 'ftp_data', 'gopher', 'harvest', 'hostnames', 'http', 'http_2784', 'http_443', 'http_8001', 'imap4', 'IRC', 'iso_tsap', 'klogin', 'kshell', 'ldap', 'link', 'login', 'mtp', 'name', 'netbios_dgm', 'netbios_ns', 'netbios_ssn', 'netstat', 'nnsp', 'nntp','ntp_u', 'other', 'pm_dump', 'pop_2', 'pop_3', 'printer', 'private', 'red_i', 'remote_job', 'rje', 'shell','smtp', 'sql_net', 'ssh', 'sunrpc', 'supdup', 'systat', 'telnet', 'tftp_u', 'tim_i', 'time', 'urh_i', 'urp_i','uucp', 'uucp_path', 'vmnet', 'whois', 'X11', 'Z39_50', 'icmp'}

@attribute 'flag' { 'OTH', 'REJ', 'RSTO', 'RSTOS0', 'RSTR', 'S0', 'S1', 'S2', 'S3', 'SF', 'SH' }

@attribute 'source_bytes' real

@attribute 'destination_bytes' real

@attribute 'land' {'0', '1'}

@attribute 'wrong- fragment' real

@attribute 'urgent' real

@attribute 'hot' real

@attribute 'num_failed_logins' real

@attribute 'logged_in' {'0', '1'}

@attribute 'num_compromised' real

@attribute 'root_shell' real

@attribute 'su_attempted' real

@attribute 'num_root' real

@attribute 'num_file_creations' real

@attribute 'num_shells' real

@attribute 'num_access_files' real

@attribute 'num_outbound_cmds' real

@attribute 'is_host_login' {'0', '1'}

@attribute 'is_guest_login' {'0', '1'}

@attribute 'count' real

@attribute 'srv_count' real

@attribute 'serror_rate' real

@attribute 'srv_serror_rate' real

@attribute 'rerror_rate' real

@attribute 'srv_rerror_rate' real

@attribute 'same_srv_rate' real

@attribute 'diff_srv_rate' real

@attribute 'srv_diff_host_rate' real

@attribute 'destination_host_count' real

@attribute 'destination_host_srv_count' real

@attribute 'destination_host_same_srv_rate' real

@attribute 'destination_host_diff_srv_rate' real

@attribute 'destination_host_same_source_port_rate' real

@attribute 'destination_host_srv_diff_host_rate' real

@attribute 'destination_host_serror_rate' real

@attribute 'destination_host_srv_serror_rate' real

@attribute 'destination_host_rerror_rate' real

@attribute 'destination_host_srv_rerror_rate' real

@attribute 'class'

This ARFF file format shows the format of dataset used in this research.total 41 features are available in data set.

Initially @ relation tag shows the name of relation file.

@ attribute tag shows name of all the attributes one by one

@ data shows data taken for research in ARFF file format.

Sample ARFF file data

@data

0,tcp,ftp_data,SF,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,150,25,0.17,0.03,0.17,0.00,0.00,0.00,0.05,0.00,normal,20

0,udp,other,SF,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,1,0.00,0.00,0.00,0.00,0.08,0.15,0.00,255,1,0.00,0.60,0.88,0.00,0.00,0.00,0.00,0.00,normal,15

0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,123,6,1.00,1.00,0.00,0.00,0.05,0.07,0.00,255,26,0.10,0.05,0.00,0.00,1.00,1.00,0.00,0.00,neptune,19

0,tcp,http,SF,232,8153,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,5,5,0.20,0.20,0.00,0.00,1.00,0.00,0.00,30,255,1.00,0.00,0.03,0.04,0.03,0.01,0.00,0.01,normal,21

0,tcp,http,SF,199,420,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,30,32,0.00,0.00,0.00,0.00,1.00,0.00,0.09,255,255,1.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,normal,21

0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,121,19,0.00,0.00,1.00,1.00,0.16,0.06,0.00,255,19,0.07,0.07,0.00,0.00,0.00,0.00,1.00,1.00,neptune,21

0,udp,domain_u,SF,44,133,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,73,75,0.00,0.00,0.00,0.00,1.00,0.00,0.03,122,212,0.88,0.02,0.88,0.01,0.00,0.00,0.08,0.00,normal

0,icmp,eco_i,SF,8,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,15,0.00,0.00,0.00,0.00,1.00,0.00,1.00,2,46,1.00,0.00,1.00,0.26,0.00,0.00,0.00,0.00,nmap

0,tcp,uucp,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,135,9,1.00,1.00,0.00,0.00,0.07,0.06,0.00,255,11,0.04,0.07,0.00,0.00,1.00,1.00,0.00,0.00,neptune

0,tcp,finger,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,24,12,1.00,1.00,0.00,0.00,0.50,0.08,0.00,255,59,0.23,0.04,0.00,0.00,1.00,1.00,0.00,0.00,neptune

Above sample data shows data value for following features.

List of features

Duration , Protocol_Type , Service , Source_Bytes , Destination_Bytes , Flag , Land , Wrong_Fragment , Urgent , Hot , Num_Failed_Logins , Logged_In , Num_Compromised , Root_Shell , Su_Attempted , Num_Root , Num_File_Creations Num_Shells , Num_Access_Files , Num_Outbound_Cmds, Is_Hot_Login , Is_Guest_Login , Count , Serror_Rate , Rerror_Rate , Same_Srv_Rate ,Diff_Srv_Rate Srv_Count , Srv_Serror_Rate , Srv_Rerror_Rate , Srv_Diff_Host_Rate, Destination_host_count, Destination_host_srv_count, Destination_host_same_srv_rate, Destination_host_diff_srv_rate, Destination_host_same_source_port_rate, Destination_host_srv_diff_ host_rate, Destination_host_serror_rate, Destination_host_srv_serror_rate, Destination_host_rerror_rate,

For experiment, weka software is used. Figure 5.2 represents weka software explorer screen,which provide facility to open files into csv format and convert it to arff file format. Weka is machine learning and data mining software weka software analyses any file based on the attributes. Weka provides explorer to analyze any file . Following figure shows explorer screen which represents the relation between any attribute and related class label. Explorer mode is chosen as it has capability to represent analysis in graphical way also.

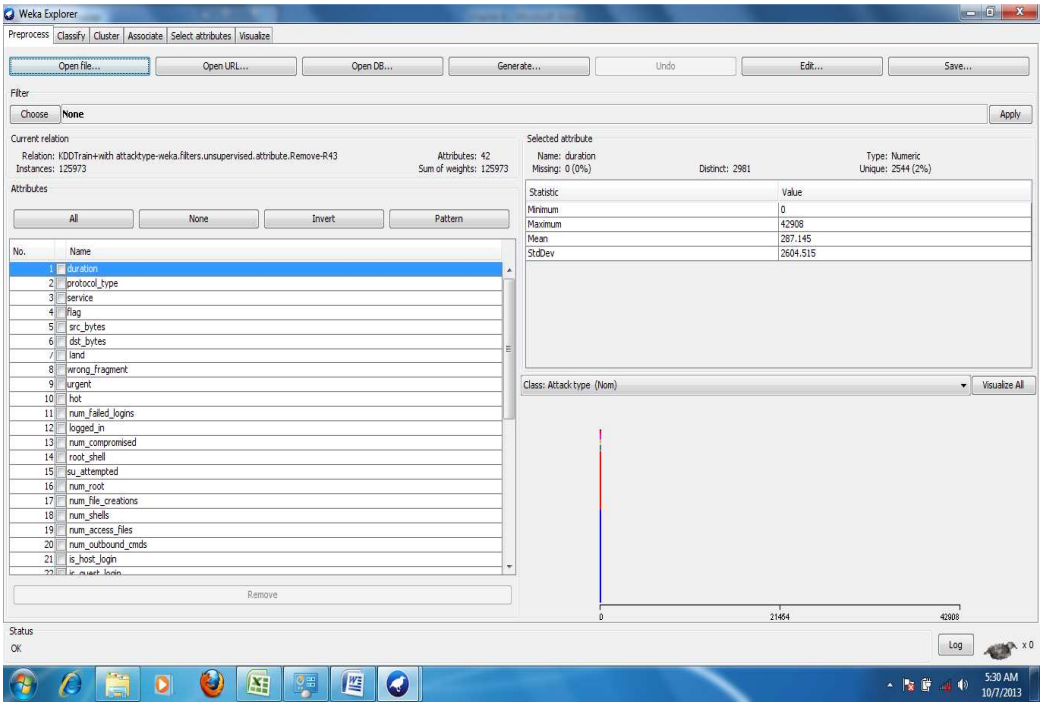


Figure 5.2 Weka Explorer Screen

5.3. Details of experiments

All experiments are performed in a computer with the configurations Intel(R) Core(TM) 2 CPU 2.16GHz, 4 GB RAM, and the operating system platform is Microsoft Windows XP. Weka is collections of machine learning algorithms for data mining tasks.

Experiments are performed in following manner.

- In the beginning, in order to execute experiments, the researcher selected the dataset, for experiment NSL KDD data set is used.
- Further, data mining software weka is used .
- The selected dataset is converted to ARFF file format .ARFF is the file format supported by Weka.
- To come up with cleaned datasets preprocessing tasks are undertaken for underling missing values, removing additional features.
- Feature selection methods are used to select the most relevant features for intrusion detection. For feature selection, there is a filtering technique which is called Attribute Selection under supervised approach or under select attribute menu. Evaluate the trained classifier using all attributes or some selected attributes by excluding unimportant features to achieve Feature ranking and selection of relevant features. After selecting either all features or some selected features develop the classifier model for different predictive modeling techniques. A supervised attribute filter that is used to select attributes, is very flexible and allows various search and evaluation methods to be combined.
- Train the classifier using WEKA data mining software.
- Here, there were a number of experiments done by changing different test options and classifier techniques. The performance comparison between different experimentation was evaluated and discussed.
- Further a training model which gives best performance for classification is selected for this study
- Lastly, the selected model for this study is tested by previously unseen records which were unlabelled that enable to determine the performance of the selected model.

Experiment no. 1:

The aim of this experiment is to investigate and evaluate the performance of J48 classification algorithm with percentage split validation method. In this experiment data is trained with J48 algorithm with default parameters.

Figure 5.3 shows sample output for J48 classification algorithm with percentage split validation method with Test mode: split 75.0% train, remainder test

```
Scheme:     weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:   KDDTrain+with attacktype-
weka.filters.unsupervised.attribute.Remove-R43-
weka.filters.supervised.attribute.AttributeSelection-
Eweka.attributeSelection.CfsSubsetEval-
Sweka.attributeSelection.BestFirst -D 1 -N 5
Instances:  125973
Attributes: 20
=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances       31408          99.7301 %
Incorrectly Classified Instances        85           0.2699 %
Kappa statistic                          0.9955
Mean absolute error                      0.0003
Root mean squared error                  0.0148
Relative absolute error                  0.6348 %
```

Figure 5.3 Performance of J48 classification algorithm with percentage split

In this experiment, J48 classifier algorithm run on a training set with 20 attributes took 53.89 seconds to build the model and the model generated tree with a J48 decision tree having 698 numbers of leaves and size of tree is 811. Total Number of examples taken for experiment is 31493 .

Experiment no. 2 :

The aim of this experiment is to investigate and evaluate the performance of J48 classification algorithm with 10 fold cross validation method. In this experiment data is trained with J48 algorithm with default parameters.

Created decision tree has having 698numbers of leaves and tree size is 811 size. Figure 5.4 shows sample output for J48 classification algorithm with 10 fold cross validation method.

```
Scheme:     weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:   KDDTrain+with attacktype-
weka.filters.unsupervised.attribute.Remove-R43-
weka.filters.supervised.attribute.AttributeSelection-
Eweka.attributeSelection.CfsSubsetEval-
Sweka.attributeSelection.BestFirst -D 1 -N 5
Instances:  125973
Attributes:  20
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      125648          99.742 %
Incorrectly Classified Instances       325          0.258 %
Kappa statistic                          0.9957
Mean absolute error                      0.0003
Root mean squared error                  0.0145
Relative absolute error                  0.656 %
```

Figure 5.4 performance ofJ48 classification algorithm with 10 fold cross validation

In this experiment, J48 classifier algorithm run on a training set with 20 attributes took 52.81 seconds to build the model and the model generated tree with a J48 decision tree having 698 numbers of leaves and 811 tree sizes with Total Number of examples taken for experiment is 31493.This experiment shows True Positive Rate 0.997 ,false positive rate is 0.002,Precision 0.997 , Recall value of experiment is 0.997 , F-Measure is 0.997 , ROC Area obtained through experiment is 0.999.

Experiment no. 3:

The aim of this experiment is to investigate and evaluate the performance of rule based classification algorithm ONE R with percentage split validation method. In this experiment data is trained with ONE R algorithm with default parameters.

Figure 5.5. shows sample output for rule based classification algorithm ONER with percentage split validation method.

```
Scheme:     weka.classifiers.rules.OneR -B 6

Relation:   KDDTrain+with attacktype-
weka.filters.unsupervised.attribute.Remove-R43-
weka.filters.supervised.attribute.AttributeSelection-
Eweka.attributeSelection.CfsSubsetEval-
Sweka.attributeSelection.BestFirst -D 1 -N 5

Instances:  125973

Attributes: 20
=== Evaluation on test split ===

=== Summary ===


Correctly Classified Instances       28570               90.7186 %
Incorrectly Classified Instances      2923                9.2814 %
Kappa statistic                          0.8443
Mean absolute error                      0.0081
Root mean squared error                  0.0898
Relative absolute error                 15.3495 %
```

Figure 5.5 Performance of ONE R classification algorithm with percentage split

In this experiment, rule based classification algorithm ONE R run on a training set with 20 attributes. True Positive Rate is 0.907, false Positive Rate is 0.042, Precision observed is 0.898 ,Recall value obtained is 0.907 with F-Measure 0.889 and ROC area obtained through experiment is. 0.933.

Experiment no. 4 :

The aim of this experiment is to investigate and evaluate the performance of rule based classification algorithm One R with 10 fold cross validation method. In this experiment data is trained with ONE R algorithm with default parameters.

Figure 5.6 shows sample output for rule based classification algorithm One R with 10 fold cross validation method.

```
Scheme:     weka.classifiers.rules.OneR -B 6

Relation:   KDDTrain+with attacktype-
weka.filters.unsupervised.attribute.Remove-R43-
weka.filters.supervised.attribute.AttributeSelection-
Eweka.attributeSelection.CfsSubsetEval-
Sweka.attributeSelection.BestFirst -D 1 -N 5

Instances:  125973

Attributes:  20
=== Stratified cross-validation ===

=== Summary ===


Correctly Classified Instances      114527          90.9139 %
Incorrectly Classified Instances     11446           9.0861 %
Kappa statistic                          0.8478
Mean absolute error                      0.0079
Root mean squared error                  0.0889
Relative absolute error                 15.0351 %
```

Figure 5.6 Performance of ONE R classification algorithm with 10 fold cross validation

In this experiment, rule based classification algorithm ONE R run on a training set with 20 attributes. True Positive Rate is 0.909, false Positive Rate is 0.039, Precision observed is 0.905,Recall value obtained is 0.909 with F-Measure 0.892 and ROC obtained through experiment is 0.935.

Experiment no. 5 :

The aim of this experiment is to investigate and evaluate the performance of Bayes net classification algorithm with percentage split validation method. The classifier uses 20 features out of the total 41 features for training classifier. Figure 5.7 shows sample output for Bayes net classification algorithm with percentage split validation method.

```
Scheme:       weka.classifiers.bayes.BayesNet -D -Q
weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E
weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

Relation:     KDDTrain+with attacktype-
weka.filters.unsupervised.attribute.Remove-R43-
weka.filters.supervised.attribute.AttributeSelection-
Eweka.attributeSelection.CfsSubsetEval-
Sweka.attributeSelection.BestFirst -D 1 -N 5

=== Evaluation on test split ===

=== Summary ===


Correctly Classified Instances       30853               97.9678 %
Incorrectly Classified Instances       640               2.0322 %
Kappa statistic                          0.9666
Mean absolute error                      0.0017
Root mean squared error                  0.0341
Relative absolute error                  3.1659 %
```

Figure 5.7 Performance of BAYES NET classification algorithm with percentage split

TP Rate observed is 0.98 , FP Rate observed is 0.002 , Precision observed is 0.983 Recall observed is 0.98 , F-Measure observed is 0.98, ROC Area obtained through experiment is 1.

Experiment no. 6:

The aim of this experiment is to investigate and evaluate the performance of Bayes net classification algorithm with 10 fold cross validation method. Figure 5.8 shows sample output for Bayes net classification algorithm with 10 fold cross validation method.

```
Scheme:       weka.classifiers.bayes.BayesNet -D -Q
weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E
weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

Relation:     KDDTrain+with attacktype-
weka.filters.unsupervised.attribute.Remove-R43-
weka.filters.supervised.attribute.AttributeSelection-
Eweka.attributeSelection.CfsSubsetEval-
Sweka.attributeSelection.BestFirst -D 1 -N 5

Instances:    125973

Attributes:   20
=== Stratified cross-validation ===

=== Summary ===


Correctly Classified Instances      123319          97.8932 %
Incorrectly Classified Instances      2654           2.1068 %
Kappa statistic                          0.9654
Mean absolute error                      0.0017
Root mean squared error                  0.0349
```

Figure 5.8 performance of BAYES NET classification algorithm with 10 fold cross validation

In this experiment, bayes net classification algorithm run on a training set with 20 attributes. True Positive Rate is 0.979, false Positive Rate is 0.003, Precision observed is 0.982 ,Recall value obtained is 0.979 with F-Measure 0.979 and ROC obtained through experiment is 1.

Experiment no. 7 :

The aim of this experiment is to investigate the effect of ensemble method on classifiers performance. Ensemble method chosen for experiment is boosting . Adaboost algorithm of boosting with 10 fold cross validation method, is taken for experiment. The classifier uses 20 features out of the total 41 features for training classifier. Figure 5.9 shows sample output for J48 with ensemble method Adaboost.

```
Scheme:weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W
weka.classifiers.trees.J48 -- -C 0.25 -M 2

Relation:     KDDTrain+with attacktype-
weka.filters.unsupervised.attribute.Remove-R43-
weka.filters.supervised.attribute.AttributeSelection-
Eweka.attributeSelection.CfsSubsetEval-
Sweka.attributeSelection.BestFirst -D 1 -N 5

Instances:    125973

Attributes:   20

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances      125790          99.8547 %
Incorrectly Classified Instances       183          0.1453 %
Kappa statistic                          0.9976
Mean absolute error                      0.0001
Root mean squared error                  0.0106
Relative absolute error                  0.2621 %
```

Figure 5.9 Performance of J48 classification algorithm with boosting

In this experiment, J48 with boosting classification algorithm run on a training set with 20 attributes. True Positive Rate is 0.999 , false Positive Rate is 0.001, Precision observed is 0.998 ,Recall value obtained is 0.999 with F-Measure 0.998 and ROC obtained through experiment is 1.

Experiment no. 8 :

The aim of this experiment is to investigate the effect of ensemble method on classifiers performance. Ensemble method chosen for experiment is Bagging algorithm with 10 fold cross validation method. The classifier uses 20 features out of the total 41 features for training classifier. Figure 5.10 shows sample output for J48 with ensemble method Bagging algorithm with 10 fold cross validation method.

```
Scheme:weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W
weka.classifiers.trees.J48 -- -C 0.25 -M 2

Relation:     KDDTrain+with attacktype-
weka.filters.unsupervised.attribute.Remove-R43-
weka.filters.supervised.attribute.AttributeSelection-
Eweka.attributeSelection.CfsSubsetEval-
Sweka.attributeSelection.BestFirst -D 1 -N 5

Instances:    125973

Attributes:   20

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances      125697          99.7809 %
Incorrectly Classified Instances       276          0.2191 %
Kappa statistic                          0.9964
Mean absolute error                      0.0003
Root mean squared error                  0.0124
Relative absolute error                  0.6426 %
```

Figure 5.10 Performance of J48 classification algorithm with bagging

In this experiment, J48 with bagging classification algorithm run on a training set with 20 attributes. True Positive Rate is 0.998 , false Positive Rate is 0.001, Precision observed is 0.998, Recall value obtained is 0.998 with F-Measure 0.998 and ROC obtained through experiment is 1.

Experiment no. 9:

The aim of this experiment is to investigate evaluate the performance of J48 classification algorithm with 10 fold cross validation method. In this experiment, attribute selection filter is not applied, therefore all the 42 attributes are used for construction of classification decision tree. Using these default parameters, the classification model is developed with a J48 decision tree having 671 numbers of leaves and 852 tree size.

```
Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     KDDTrain+with attacktype-
weka.filters.unsupervised.attribute.Remove-R43
Instances:    125973
Attributes:   42
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances      125666               99.7563 %
Incorrectly Classified Instances       307               0.2437 %
Kappa statistic                          0.996
Mean absolute error                      0.0003
Root mean squared error                  0.0141
Relative absolute error                  0.5621 %
```

Figure 5.11 Performance ofJ48 classification algorithm without attribute selection

In this experiment, J48 without feature selection algorithm run on a training set with 42 attributes. True Positive Rate is 0.998 , false Positive Rate is 0.002, Precision observed is 0.997 ,Recall value obtained is 0.998 with F-Measure 0.997 and ROC obtained through experiment is 0.999.

Experiment no.10 :

Data discretization is a procedure that takes a data set and converts all continuous attributes to categorical. Supervised discretization method is used here since majority of datasets contains class labels.

```
Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2

Relation:     KDDTrain+with attacktype-
weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-Rfirst-
last-weka.filters.unsupervised.attribute.Remove-R43-
weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-Rfirst-
last

Instances:    125973

Attributes:   42


=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances      124443          98.7855 %

Incorrectly Classified Instances     1530           1.2145 %

Kappa statistic                          0.9798

Mean absolute error                      0.0018

Root mean squared error                  0.0312

Relative absolute error                  3.459  %
```

Figure 5.12 Performance ofJ48 classification algorithm with filter discritization.

In this experiment, J48 with filter discritization is used classification algorithm run on a training set with 20 attributes. True Positive Rate is 0.988 , false Positive Rate is 0.011, Precision observed is 0.986 ,Recall value obtained is 0.988 with F-Measure 0.986 and ROC obtained through experiment is 994.

5.4. Comparison of classifiers

Classifiers are compared on performance measurement terms like Correctly classified instance, Incorrectly classified instance these two measures represents how many records are classified correctly and how many records are not. Kappa statistics is also used for comparison of classifiers. Mean absolute error, Relative_absolute_error, Root mean squared error, Root relative squared error are the other terms on which classifiers are compared. Initially J48, OneR, Bayes net classifiers are compared for evaluation of performance using 10 fold cross validation.

Comparison of 10 experiments on the basis of True positive rate is shown in table 6.1.

Table 5.2 Comparison of 10 experiments on the basis of True positive rate

Experiment number	Name Of Experiment	TP Rate
Experiment1	J48 Percentage Split	0.997
Experiment2	J48 10 Fold	0.997
Experiment3	One R Percentage Split	0.907
Experiment4	One R 10 Fold	0.909
Experiment5	Bayes Percentage Split	0.98
Experiment6	Bayes 10 Fold	0.978
Experiment7	J48-Adaboost 10 Fold	0.999
Experiment8	J48-Bagging 10 Fold	0.998
Experiment9	J48 Discritize	0.988
Experiment10	J48 Without Attribute Selection	0.998

Chart 5.1 shows comparison of 10 experiment on true positive rate. This Chart clearly shows performance of One R, bayes net with 10 fold cross validation, percentage split is significantly lower than J48 algorithm .

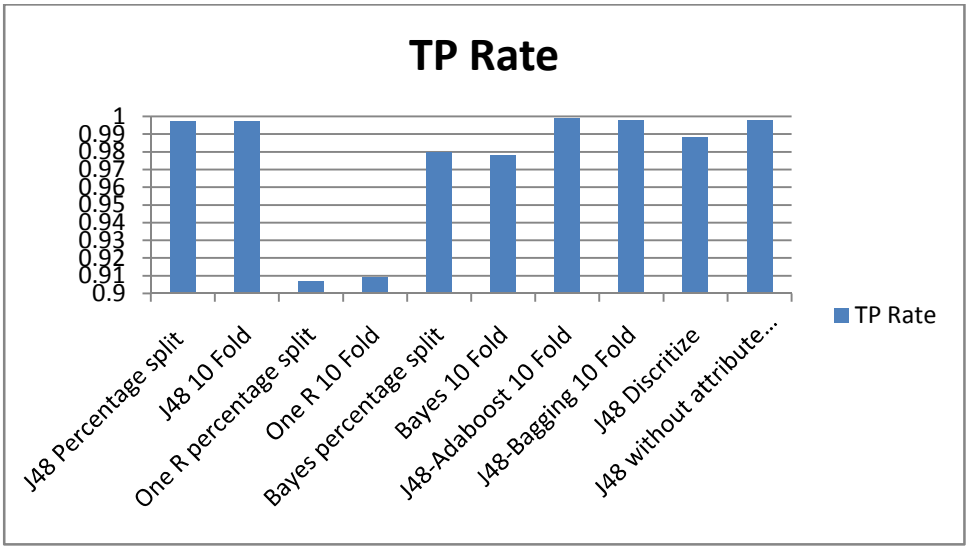


Chart 5.1 Comparison of all experiments on TP Rate

Comparison of 10 experiments on the basis of False positive rate is shown in table 6.2.

Table 5.3 Comparison of 10 experiments on the basis of False positive rate

Experiment number	Name Of Experiment	FP Rate
Experiment1	J48 Percentage Split	0.002
Experiment2	J48 10 Fold	0.002
Experiment3	One R Percentage Split	0.042
Experiment4	One R 10 Fold	0.039
Experiment5	Bayes Percentage Split	0.002
Experiment6	Bayes 10 Fold	0.003
Experiment7	J48-Adaboost 10 Fold	0.001
Experiment8	J48-Bagging 10 Fold	0.001
Experiment9	J48 Discritize	0.011
Experiment10	J48 Without Att Selection	0.002

Chart 5.2 shows comparison of 10 experiments on false positive rate. This Chart clearly shows performance of One R is lower than other algorithm. Chart shows

high false positive rate means if incoming data is not attack data but algorithm indicates it is attack data.

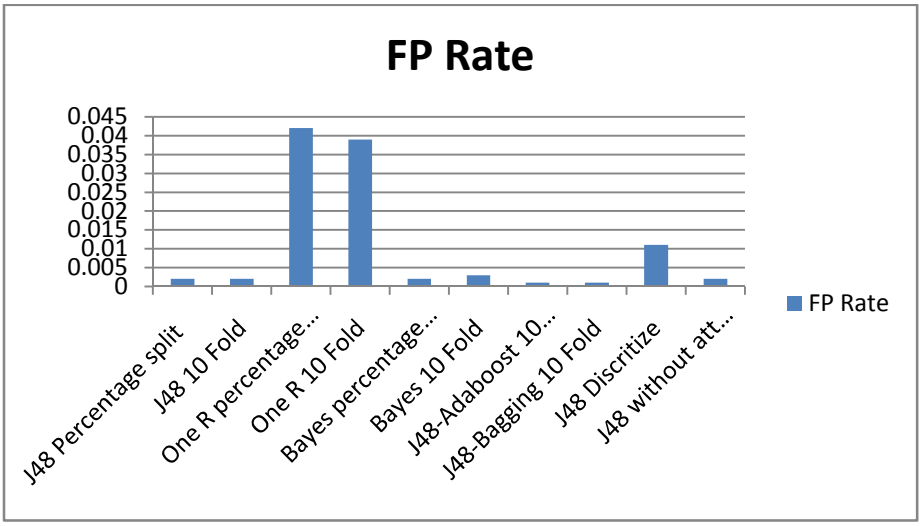


Chart 5.2 Comparison of all experiments on FP Rate

Comparison of 10 experiments on the basis of Time taken to build model is shown in table 6.3.

Table 5.4 Comparison of 10 experiments on the basis of Time taken

Experiment number	Name Of Experiment	Time Taken
Experiment1	J48 Percentage Split	53.89
Experiment2	J48 10 Fold	52.81
Experiment3	One R Percentage Split	5.24
Experiment4	One R 10 Fold	4.19
Experiment5	Bayes Percentage Split	9.17
Experiment6	Bayes 10 Fold	9.09
Experiment7	J48-Adaboost 10 Fold	451.52
Experiment8	J48-Bagging 10 Fold	489.81
Experiment9	J48 Discritize	94.66
Experiment10	J48 Without Att Selection	147.53

Chart 5.3 shows comparison of 10 experiments for time taken to classify. This Chart evidently shows performance of ensemble methods like bagging and

boosting taken long time for classification as compared to other algorithms. Whereas J48 algorithm ,One R algorithm takes less time.

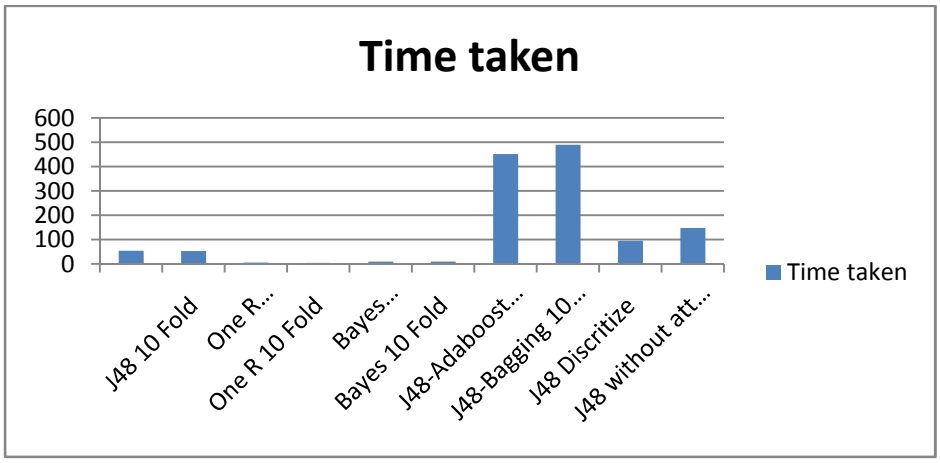


Chart 5.3 Comparison of all experiments on time taken

Comparison of 10 experiments on the basis of correctly classified instances is shown in table 64.4.

Table 5.5 Comparison of 10 experiments on the basis of correctly classified instances

Experiment number	Name Of Experiment	Correctly Classified
Experiment1	J48 Percentage Split	99.7301
Experiment2	J48 10 Fold	99.742
Experiment3	One R Percentage Split	90.7186
Experiment4	One R 10 Fold	90.9139
Experiment5	Bayes Percentage Split	97.9678
Experiment6	Bayes 10 Fold	97.3732
Experiment7	J48-Adaboost 10 Fold	99.8547
Experiment8	J48-Bagging 10 Fold	99.7809
Experiment9	J48 Discritize	99.7079
Experiment10	J48 Without Att Selection	99.7563

Chart 5.4 shows comparison of 10 experiments on correctly classified instances. This Chart clearly shows performance of One R is lower than other algorithm.

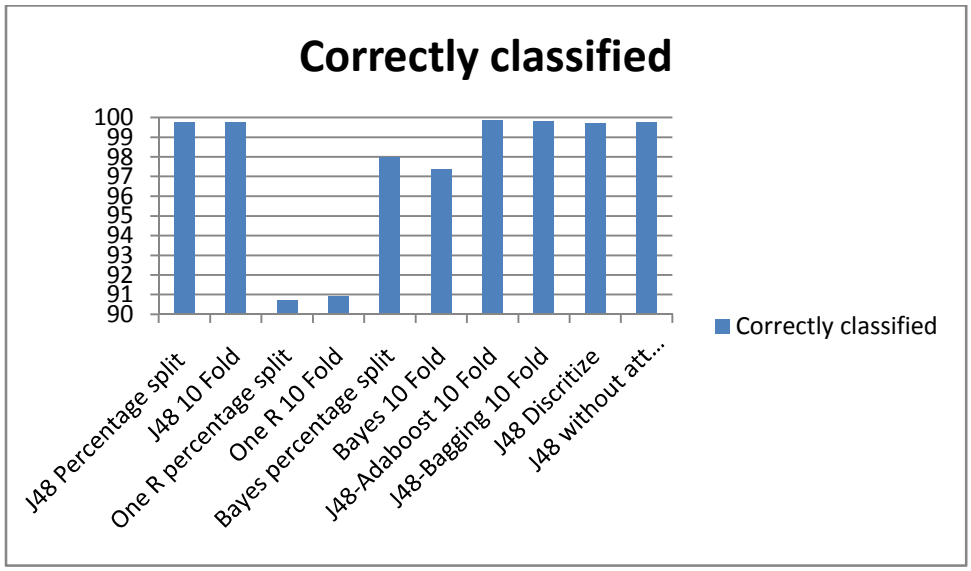


Chart 5.4 Comparison of all experiments on correctly classified instances

To analyze all the experiment their performance is evaluated based on percentage split. in table 4.5 J48(decision tree), One R (rule based) ad bayes net (bayes based) are compared on the basis of Correctly classified instance, Incorrectly classified instance these two measures represents how many records are classified correctly and how many records are not. Kappa statistics is also used for comparison of classifiers. Mean absolute error, Relative_absolute_error, Root mean squared error, Root relative squared error are the other terms on which classifiers are compared

Table 5.6 : Comparison Of J48 Algorithm With Other classification Algorithms

Sr.no.	**Parameter**	**J48**	**OneR**	**Bayes Net**
1	Correctly classified instance	99.742	90.9139	97.3732
2	Incorrectly classified instance	0.258	9.0861	2.6268
3	Kappa statistics	0.9957	0.8478	0.9571
4	Mean absolute error	0.0003	0.0079	0.0024
5	Root mean squared error	0.0145	0.0889	0.0434
6	Relative_absolute_error	0.656	15.0351	4.5617
7	Root relative squared error	8.9515	54.8395	26.7982

The table 6.5 shows comparison of three classification concepts like decision tree, rule based algorithm , here J48 belong to decision tree ,ONE R belongs to rule

based classifier and bayes net represents bayes classification . For comparison ,10 fold cross validation is used . From this comparison it is clearly visible that Decision tree J48 perform better than ONE R and bayes net.

Chart 5.5 represents comparison of 3 types classifiers for correctly classified instance. This Chart clearly shows performance of J48 algorithm is far better than other algorithms.

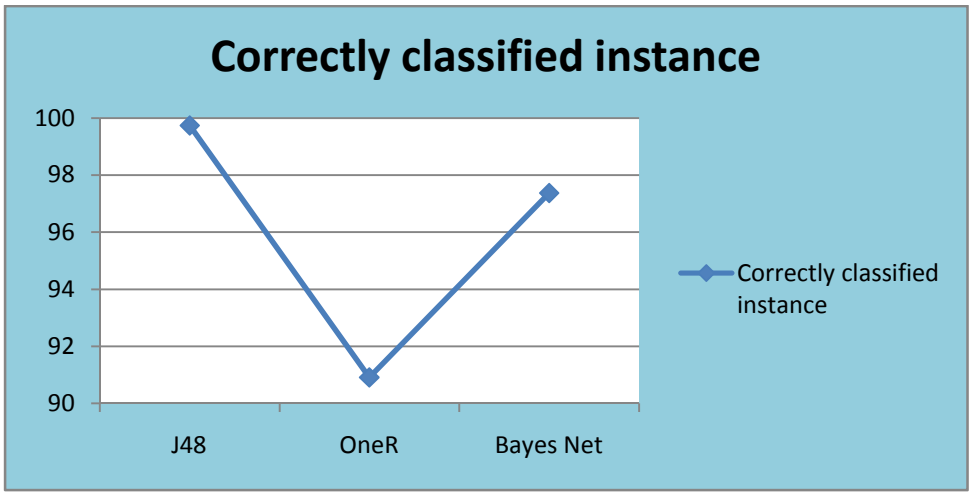


Chart 5.5 Comparison of 3 types classifiers for correctly classified instance

Chart 5.6 represent comparison of classifiers for relative absolute error. Chart represents comparison of 3 types classifiers for relative absolute error. Error rate is low in J48 algorithm whereas One R and bayes algorithm have high error rate. This Chart clearly shows performance of J48 algorithm is far better than other algorithms as it has low error rate.

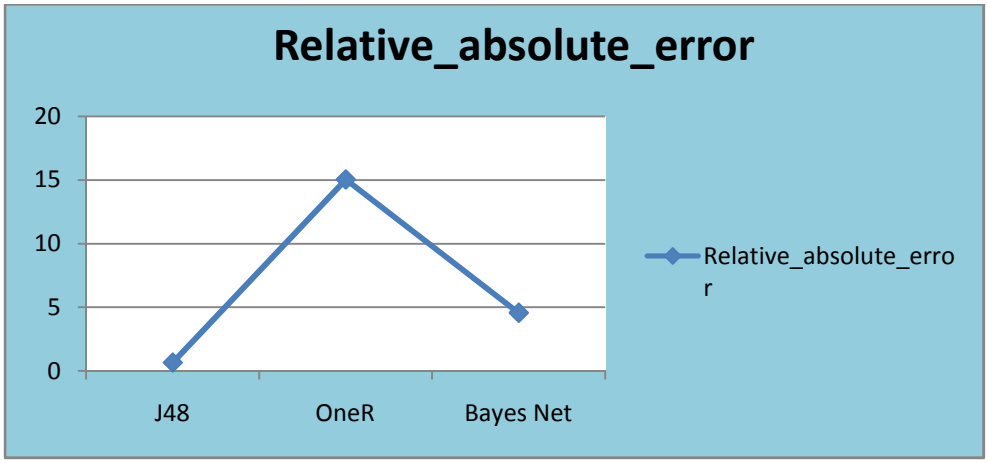


Chart 5.6 Comparison of 3 types of classifiers for relative absolute error

Classifiers are compared on performance measurement terms like Correctly classified instance, Incorrectly classified instance . These two measures represent how many records are classified correctly and how many records are not. Kappa statistics is also used for comparison of classifiers. Mean absolute error, Relative_absolute_error, Root mean squared error, Root relative squared error are the other terms on which classifiers are comparedInitially J48, OneR, Bayes net classifiers are compared for evaluation of performance using 10 fold cross validation.

Table 5.7 : Comparison of classification algorithm using percentage split .

Sr.no.	**Parameter**	**J48**	**OneR**	**Bayes Net**
1	Correctly classified instance	99.7301	90.7186	97.9678
2	Incorrectly classified instance	0.2699	9.2814	2.0322
3	Kappa statistics	0.9955	0.8443	0.9666
4	Mean absolute error	0.0003	0.0081	0.0017
5	Root mean squared error	0.0148	0.0898	0.0434
6	Relative_absolute_error	0.634	15.3495	3.1659
7	Root relative squared error	9.1177	55.3802	21.0098

The table 6.6 shows comparison of three classification concepts like decision tree, rule based algorithm , here J48 belongs to decision tree ,ONE R belongs to rule based classifier and bayes net represents bayes classification . For comparison, percentage

split validation is used . From this comparison, it is clearly visible that Decision tree J48 perform better than ONE R and bayes net.

Table 5.8 : Comparison of J48 Algorithm With And Without Feature Selection

Sr.no.	Parameter	J48 without feature selection	J48 with feature selection	J48 With discritizati-on
1	Correctly classified instance	99.7563	99.742	99.7079
2	Incorrectly classified instance	0.2437	0.258	0.2921
3	Kappa statistics	0.996	0.9957	0.9952
4	Mean absolute error	0.0003	0.0003	0.0004
5	Root mean squared error	0.0141	0.0145	0.0149
6	Relative_absolute_error	0.5621	0.656	0.7006

5.

The table 6.7 shows comparison of classifier with three different strategies like with attribute selection, without attribute selection and discritization. For comparison, 10 fold cross validation is used. From this comparison it is clearly visible that Decision tree J48 performs better then attribute selection is used.

Table 5.9 : Comparison of time taken by J48 algorithm with and without feature selection

Sr.no.	Parameter	J48 without feature selection	J48 with feature selection	J48 With discritization
1	Time taken to build model:	147.53 seconds	50.74seconds	94.66 seconds

The table 6.8 shows comparison of time taken by classifier with three different strategies like with attribute selection, without attribute selection and discritization. For comparison, 10 fold cross validation is used. From this comparison, it is clearly visible that Decision tree J48 performs better then attribute selection is used.

Table 5.10 : Comparison of accuracy of classifiers with and without ensemble methods

Sr.no.	**Parameter**	**J48**	**J48 With bagging**	**J48 With Boosting**
1	Correctly classified instance	99.742	99.7809	99.8547
2	Incorrectly classified instance	0.258	0.2191	0.1453
3	Kappa statistics	0.9957	0.9964	0.9976
4	Mean absolute error	0.0003	0.0003	0.0001
5	Root mean squared error	0.0145	0.0124	0.0106
6	Relative_absolute_error	0.656	0.6426	0.2621

The table 6.9 shows comparison of classifier without use of ensemble method. Ensemble method like bagging and boosting are used with decision tree J48 classifier. From this comparison, it is clearly visible that Decision tree J48 when used with ensemble method there is no major change in performance.

Table 5.11 : Comparison of time taken by classifiers with and without ensemble methods

Sr.no.	**Parameter**	**J48**	**J48 With bagging**	**J48 With Boosting**
1	Time taken to build model:	50.74seconds	489.81 seconds	451.52 seconds

The table 6.10 shows comparison of time taken to build classifier with and without use of ensemble method. Ensemble method like bagging and boosting are used with decision tree J48 classifier. From this comparison it is clearly visible that Decision tree J48 when used with ensemble method it take long time to build classifier.

5.5. Results of experiments

In this research work attempt has been made to analyze data mining supervised techniques intrusion detection. Series of experiments are performed is as follows

1. J48 Percentage Split .
2. J48 10 Fold
3. One R Percentage Split
4. One R 10 Fold
5. Bayes Percentage Split
6. Bayes 10 Fold
7. J48-Adaboost 10 Fold
8. J48-Bagging 10 Fold
9. J48 Discritize
10. J48 Without Attribute Selection

Comparison of all above experiment result shows that, the classifier with 10-fold cross validation using the J48 decision tree algorithm with the default parameter values showed the best classification accuracy. This classifier model has a prediction accuracy of 99.742% on the training datasets. The findings of this study have shown that the data mining methods generate interesting rules that are crucial for intrusion detection and prevention in the networking industry.

For building a data mining model for intrusion detection, J48 decision tree algorithm with feature selection and without ensemble method gives best performance as per performance measurement terms mentioned. Based on the experiments results SIDDM model is developed.

5.6. SIDDM model :

(Data mining framework for intrusion detection)

SIDDM (Systematic Intrusion Detection using Data Mining) model is developed in this research work. Computer network security always demands, improved methods for intrusion detection. IDS are to detect all intrusions at first effectively. This demand can be fulfilled using data mining which uses intelligence technique and machine learning for detection of intrusion. These techniques are used as an alternative to expensive and strenuous human input.

In this research work a data mining model SIDDM is provided for intrusion attack classification. This model provides following

- Intrusion attacks are classified using SIDDM Model
- The output from the classifier, a set of classification rules, is used to recognize intrusion attack.
- Rules generated by SIDDM can be integrated into intrusion detection tools like Snort etc., even firewalls and detection scripts can integrate these rules to identify intrusion attack.

  Framework is constructed using following steps

- STEP 1:

  For Constructing training model dataset is taken from NSL KDD training dataset. KDD data set holds collection of network data; this data set is available on line.

- STEP 2:

  On this dataset data Preprocessing is done; in which missing values are replaced with mean of data values and removing unnecessary attributes from data set.

- STEP 3:

On preprocessed dataset, feature selection (supervised attribute selection) method is applied to select only most relevant feature which best discriminates the given class from the others. This generates reduced data set.

- STEP 4:

  On reduced dataset validation methods, 10 fold cross validation is applied.

- STEP 5:

  Decision tree classification (supervised data mining technique) is used for classification of data.

- STEP 6:

  Training of model is done with J48 decision tree classifier.

- STEP 7:

  Rules are generated for incorporating in the intrusion detection system to device the process for intrusion detection.

- STEP 8:

  For testing model KDD test dataset is used

- STEP 9:

  Model testing is done with KDD test dataset

- STEP 10:

  Predictions are made about data based on classification rules

- STEP 11:

  Attacks are classified as per classification rules

  Finally this framework is capable of Intrusion detection.

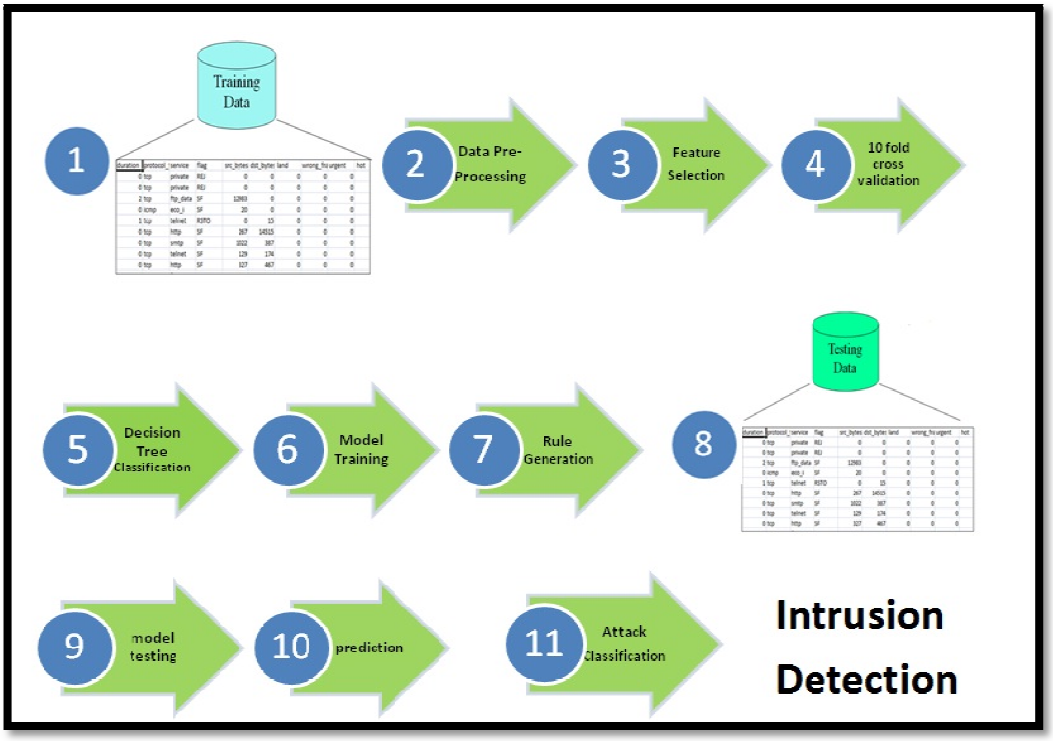


Figure 5.13 SIDDM Framework

Detail methodology used is as follows

5.6.1. Feature selection

To proceed with the building framework, feature subset selection is performed. Feature selection is the process of removing features from the data set that are irrelevant with respect to the task that is to be performed. For reduction of dimensionality feature selection methods are used. This also reduces execution time and improve predictive accuracy. In general, feature selection techniques can be categorized into two: filter methods and wrapper methods. In this research work, filter method is used .supervised attribute selection Filter methods is used for feature selection.

As can be seen from figure 5.14 twenty out of forty one features are selected. Selected features are listed below

- protocol_type
- service

- flag
- source_bytes
- Destination_bytes
- land
- wrong_fragment
- hot
- logged_in
- count
- serror_rate
- same_srv_rate
- diff_srv_rate
- Destination_host_diff_srv_rate
- Destination_host_same_source_port_rate
- Destination_host_srv_diff_host_rate
- Destination_host_serror_rate
- Destination_host_srv_serror_rate
- Destination_host_rerror_rate
- Attack type

Figure 5.14 shows weka software screen where feature selection is applied using supervised attribute selection.

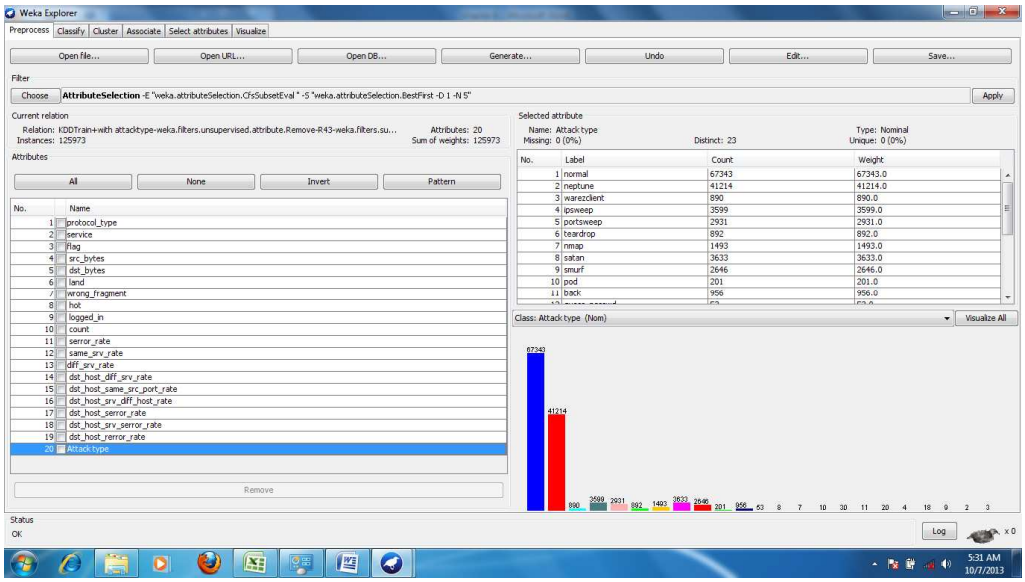


Figure 5.14 Feature Selection

Supervised data mining techniques

Supervised learning algorithms are those used in classification and prediction. If data is available with known output then these methods are useful. These training data are the data from which the classification or prediction algorithm "learns," or is "trained," about the relationship between predictor variables and the outcome variable. Supervised learning algorithm learn from the training data, it is then used to classify another new data where the outcome is not known. These algorithms can correctly classify new data.

5.6.2. Training classifier Model

Classification is task of learning a taget fuction *f* that maps each attribute set x to one of the predefined label y. this target funtion is informally known as training model. This classification model is used for predicting class label of unknown record.

- Based on the class label attribute every tuple is assumed to belong to a predefined class.
- Training set is that set of data which is used for construction of model .
- This model have decision tree or classification rules.

Figure 5.15s represents process of training classification model. For training of model at first data set with class label is considered training data. On the training data classification algorithm is applied;which builds classifier model; classifier model consists of rules.

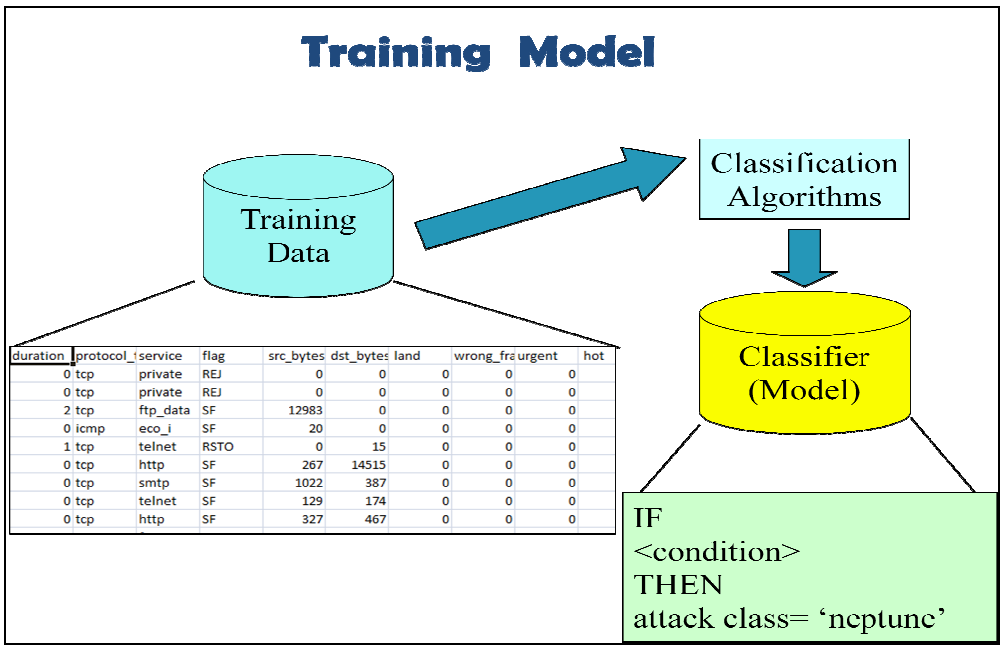


Figure 5.15 Classification Model Training

For building a data mining model for intrusion detection, J48 decision tree algorithm with feature selection and without ensemble method is used . This training model is stored and reevaluated on test set. This training model generates rules which are stored in the form

This Framework proposes following algorithm for training model. Input to this algorithm is KDD training dataset available online for research purpose. Classifier is J48 which is derived from c4.5 algorithm.

5.6.3. Proposed algorithm

Input:

Training data set (T)

Classifier:

J48

Output

Decision tree (D)

Set of Rules for intrusion detection(R).

Model (M)

Algorithm

Step 1: Preprocess the training dataset (T).

- Convert data set to ARFF file format
- Remove unnecessary attribute
- Apply supervised attribute selection filter to obtain reduced data set.

Step 2: Apply 10 fold cross validation

Step 3: perform classification using classifier j48 with parameter setting

Step 4: Generate decision tree(D) and set of rules(R)

Step 5: Generate model model(M)

This research uses KDD dataset for training model .this dataset has approx 5 million records. Data is available for 41 features related to network data. Following is list of features , separated by comma.

List of features

Duration , Protocol_Type , Service , Source_Bytes , Destination_Bytes , Flag , Land , Wrong_Fragment , Urgent , Hot , Num_Failed_Logins , Logged_In , Num_Compromised , Root_Shell , Su_Attempted , Num_Root , Num_File_Creations Num_Shells , Num_Access_Files , Num_Outbound_Cmds, Is_Hot_Login , Is_Guest_Login , Count , Serror_Rate , Rerror_Rate , Same_Srv_Rate ,Diff_Srv_Rate Srv_Count , Srv_Serror_Rate , Srv_Rerror_Rate , Srv_Diff_Host_Rate, Destination_host _count, Destination_host_srv_count, Destination_host_same_ srv_rate, Destination_host_diff_srv_rate, Destination_host_same_source_port_rate, Destination_host_srv_diff_host_rate, Destination_host_serror_rate, Destination_host _srv_serror _ rate, Destination_host_rerror_rate, Destination_host_srv_rerror_rate.

Sample of Training data used in research work

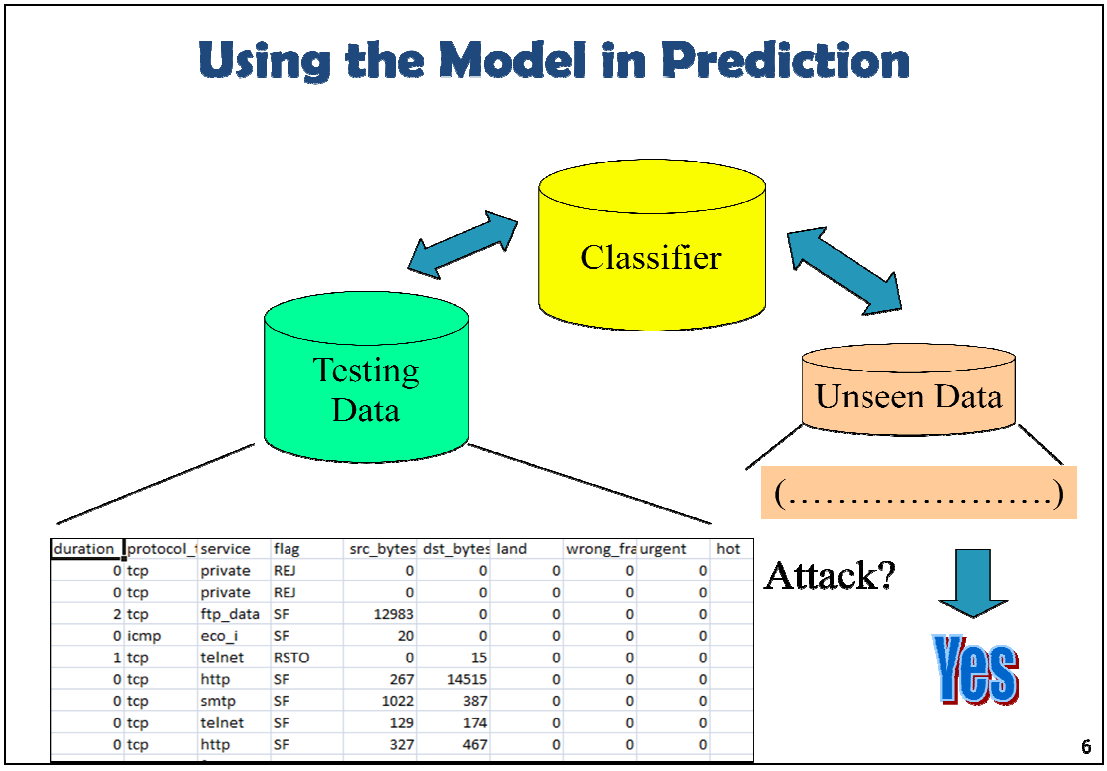
Instance 1.
0,tcp,ftp_data,SF,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,150,25,0.17,0.03,0.17,0.00,0.00,0.00,0.05,0.00,normal
Instance 2.
0,udp,other,SF,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,1,0.00,0.00,0.00,0.00,0.08,0.15,0.00,255,1,0.00,0.60,0.88,0.00,0.00,0.00,0.00,0.00,normal
Instance 3.
0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,123,6,1.00,1.00,0.00,0.00,0.05,0.07,0.00,255,26,0.10,0.05,0.00,0.00,1.00,1.00,0.00,0.00,neptune
Instance 4.
0,tcp,http,SF,232,8153,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,5,5,0.20,0.20,0.00,0.00,1.00,0.00,0.00,30,255,1.00,0.00,0.03,0.04,0.03,0.01,0.00,0.01,normal
Instance 5.

0,tcp,http,SF,199,420,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,30,32,0.00,0.00,0.00,0.00,1.00,0.00,0
.09,255,255,1.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,normal
Instance 6.
0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,121,19,0.00,0.00,1.00,1.00,0.16,0.06,
0.00,255,19,0.07,0.07,0.00,0.00,0.00,0.00,1.00,1.00,neptune
Instance 7.
0,tcp,ftp_data,SF,334,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.
00,2,20,1.00,0.00,1.00,0.20,0.00,0.00,0.00,0.00,warezclient
Instance 8.
0,tcp,name,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,233,1,1.00,1.00,0.00,0.00,0.00,0.06,0.00,
255,1,0.00,0.07,0.00,0.00,1.00,1.00,0.00,0.00,neptune
Instance 9.
0,tcp,netbios_ns,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,96,16,1.00,1.00,0.00,0.00,0.17,0.05,
0.00,255,2,0.01,0.06,0.00,0.00,1.00,1.00,0.00,0.00,neptune
Instance 10.
0,tcp,http,SF,300,13788,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,9,0.00,0.11,0.00,0.00,1.00,0.00,0
.22,91,255,1.00,0.00,0.01,0.02,0.00,0.00,0.00,0.00,normal
Instance 11.
0,icmp,eco_i,SF,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00
,1,16,1.00,0.00,1.00,1.00,0.00,0.00,0.00,0.00,ipsweep
Instance 12.
0,tcp,http,SF,233,616,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,3,3,0.00,0.00,0.00,0.00,1.00,0.00,0.0
0,66,255,1.00,0.00,0.02,0.03,0.00,0.00,0.02,0.00,normal
Instance 13.
0,tcp,http,SF,343,1178,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,9,10,0.00,0.00,0.00,0.00,1.00,0.00,0
.20,157,255,1.00,0.00,0.01,0.04,0.00,0.00,0.00,0.00,normal
Instance 14.
0,tcp,mtp,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,223,23,1.00,1.00,0.00,0.00,0.10,0.05,0.00,
255,23,0.09,0.05,0.00,0.00,1.00,1.00,0.00,0.00,neptune
Instance 15.
0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,280,17,1.00,1.00,0.00,0.00,0.06,0.05,0.
00,238,17,0.07,0.06,0.00,0.00,0.99,1.00,0.00,0.00,neptune
Instance 16.
0,tcp,http,SF,253,11905,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,10,0.00,0.00,0.00,0.00,1.00,0.00,
0.20,87,255,1.00,0.00,0.01,0.02,0.00,0.00,0.00,0.00,normal
Instance 17.
5607,udp,other,SF,147,105,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.0
0,0.00,255,1,0.00,0.85,1.00,0.00,0.00,0.00,0.00,0.00,normal
Instance 18.
0,udp,private,SF,28,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.0
0,255,2,0.01,0.02,0.01,0.00,0.00,0.00,0.77,0.00,teardrop
Instance 19.
0,tcp,http,SF,220,1398,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,26,42,0.00,0.00,0.00,0.00,1.00,0.00,
0.05,26,255,1.00,0.00,0.04,0.03,0.00,0.00,0.00,0.00,normal
Instance 20.
0,udp,domain_u,SF,43,69,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,120,120,0.00,0.00,0.00,0.00,1.00,
0.00,0.00,255,245,0.96,0.01,0.01,0.00,0.00,0.00,0.00,0.00,normal
Instance 21.0,udp,domain_u,SF,44,133,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,73,75,0.00,0.00,0.0
0,0.00,1.00,0.00,0.03,122,212,0.88,0.02,0.88,0.01,0.00,0.00,0.08,0.00,normal
Instance 22.0,icmp,eco_i,SF,8,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,15,0.00,0.00,0.00,0.00,1.
00,0.00,1.00,2,46,1.00,0.00,1.00,0.26,0.00,0.00,0.00,0.00,nmap
Instance 23.0,tcp,uucp,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,135,9,1.00,1.00,0.00,0.00,0.0
7,0.06,0.00,255,11,0.04,0.07,0.00,0.00,1.00,1.00,0.00,0.00,neptune
Instance 24.0,tcp,finger,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,24,12,1.00,1.00,0.00,0.00,0.
50,0.08,0.00,255,59,0.23,0.04,0.00,0.00,1.00,1.00,0.00,0.00,neptune

Instance 25.0,udp,domain_u,SF,43,43,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,148,228,0.00,0.00,0.00,0.00,1.00,0.00,0.01,255,255,1.00,0.00,0.01,0.00,0.00,0.00,0.00,0.00,normal

5.2.4. Testing model

Training set is used to build model, which is subsequently applied to test set, which consists of records with unknown class label. In this research NSL-KDD test is used without class label.



duration	protocol_t	service	flag	src_bytes	dst_bytes	land	wrong_fra	urgent	hot
0	tcp	private	REJ	0	0	0	0	0	
0	tcp	private	REJ	0	0	0	0	0	
2	tcp	ftp_data	SF	12983	0	0	0	0	
0	icmp	eco_i	SF	20	0	0	0	0	
1	tcp	telnet	RSTO	0	15	0	0	0	
0	tcp	http	SF	267	14515	0	0	0	
0	tcp	smtp	SF	1022	387	0	0	0	
0	tcp	telnet	SF	129	174	0	0	0	
0	tcp	http	SF	327	467	0	0	0	

Figure 5.16 Classification Model testing

This Framework proposes following algorithm for testing model. Input to this algorithm is KDD test dataset available online for research purpose. Another input to this algorithm is model (M) developed in training model phase . this algorithm make prediction for test data or unseen data whether it is normal or attack.

Algorithm proposed by SIDDM for testing /using model

Input

Test data set (E)

Model (M)

Output

Prediction for test data (P)

Classification of Intrusion detection (I).

Algorithm

1. Preprocess dataset (E)
   a. Convert data set to ARFF file format.
   b. Remove unnecessary attribute
   c. Attack field must have “?”, so now data is without label.
2. Apply model(M),this model is constructed in training phse.
3. Use test data set for testing model.
4. Make prediction(P) whether test data has attack data or normal .

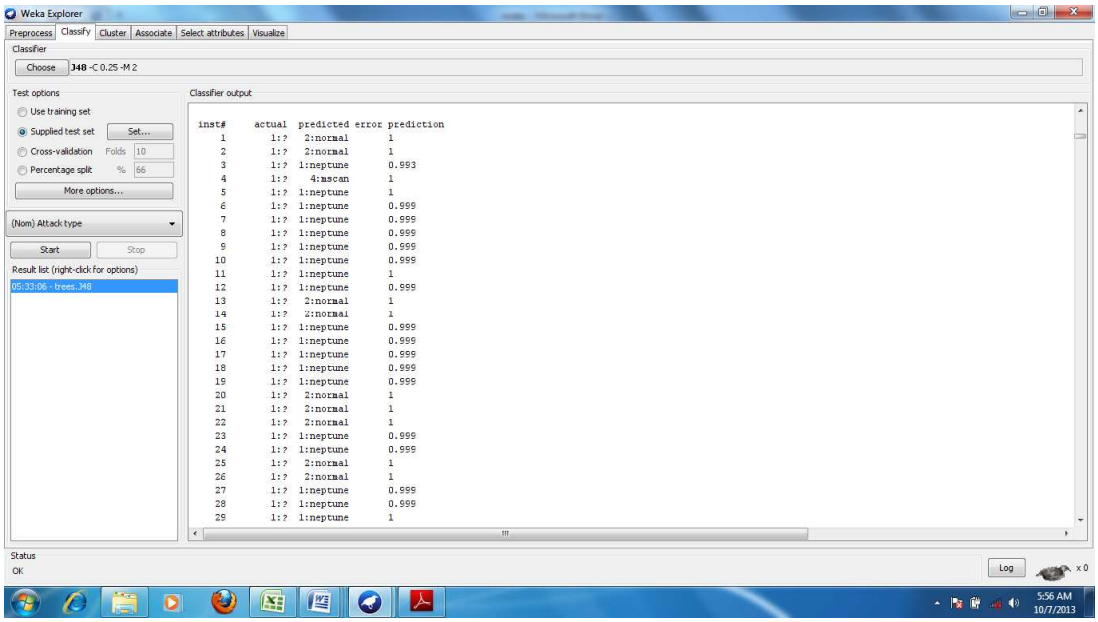


Figure 5.17 Prediction using model

In this section, the selected models from those 10 experiments conducted in this study are evaluated. From all the experiments in this study, one model has achieved better classification performance as discussed before from those experiments conducted in supervised approach; the J48 decision tree algorithm with the 10-fold cross validation model gives a better classification accuracy of predicting newly arriving intrusions in their respective class category. Prediction accuracy on the test set is 0.9999.

Following is the sample data used for testing model

```
tcp, private, rej, 0, 0, 0, 0, 0, 0, 229, 0, 0.04, 0.06, 0.06, 0, 0, 0, 0, 1, ?
tcp, private, rej, 0, 0, 0, 0, 0, 0,        136, 0, 0.01, 0.06, 0.06, 0, 0, 0, 0, 1, ?
tcp, ftp_data, sf, 12983, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0.04, 0.61, 0.02, 0, 0, 0, ?
udp, domain_u, SF, 43, 43, 0, 0, 0, 0, 111, 0, 1, 0, 0, 0, 0, 0, 0, 0, ?
tcp, private, rej, 0, 0, 0, 0, 0, 0, 483, 0.05, 0, 1, 1, 0, 0, 0, 0, 0.96, ?
icmp, ecr_i, sf, 1480, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0.52, 0, 0, 0, ?
tcp, private, rej, 0, 0, 0, 0, 0, 0, 235, 0, 0.04, 0.06, 0.07, 0, 0, 0, 0, 1, ?
tcp, private, s0, 0, 0, 0, 0, 0, 0, 206, 0.76, 0.03, 0.07, 0.08, 0, 0, 0.79, 0.29  ,0.21, ?
```

Figure 5.18 Data For Testing Model

Following table shows prediction made by SIDDM after providing network data mentioned above.

Table 5.12: Prediction made by SIDDM framework

Instance	**Actual**	**Predicted**	**Prediction Accuracy**
1	tcp, private, rej, 0, 0, 0, 0, 0, 0, 229, 0, 0.04, 0.06, 0.06, 0, 0, 0, 0, 1, ?	Normal	1
2	tcp, private, rej, 0, 0, 0, 0, 0, 0, 136, 0, 0.01, 0.06, 0.06, 0, 0, 0, 0, 1, ?	Normal	1
3	tcp, ftp_data, sf, 12983, 0, 0, 0, 0, 0,	Neptune	.999

	1, 0, 1, 0, 0.04, 0.61, 0.02, 0, 0, 0, ?		
4	udp, domain_u, SF, 43, 43, 0, 0, 0, 0, 111, 0, 1, 0, 0, 0, 0, 0, 0, 0, ?	Neptune attack	.999
5	tcp, private, rej, 0, 0, 0, 0, 0, 0, 483, 0.05, 0, 1, 1, 0, 0, 0, 0, 0.96, ?	Satan attack	1
6	icmp, ecr_i, sf, 1480, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0.52, 0, 0, 0, ?	Back attack	1
7	tcp, private, rej, 0, 0, 0, 0, 0, 0, 235, 0, 0.04, 0.06, 0.07, 0, 0, 0, 0, 1, ?	Normal	.999
8	tcp, private, s0, 0, 0, 0, 0, 0, 0, 206, 0.76, 0.03, 0.07, 0.08, 0, 0, 0.79, 0.29 , 0.21, ?	Normal	.999

5.2.5. Steps to use of framework

Intrusion detection system using the concept of SIDDM framework uses following steps to identify whether the network data is normal or attack. Figure 5.19 represents the process.

1. Collect network data.
2. Apply data preprocessing and feature selection.
3. Classification of network data based on rules generated by model.
4. Prediction whether data is attack or not.
5. Attack identification.

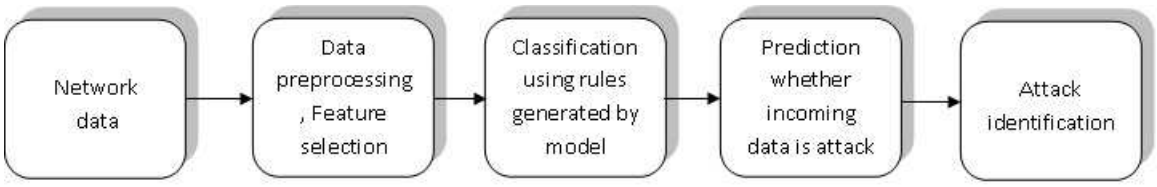


figure 5.19 Steps to use SIDDM framework

5.2.6. Rules generated

Some of the rules generated from the selected model are the following. In detail all the rules in the form of decision tree in mentioned in annexure 2.

- **RULE 1:** If protocol_type = tcp and count <= 2 and Destination_host_srv_diff_host_rate <= 0.48 and Destination_host_diff_srv_rate <= 0.1 and Destination_host_serror_rate <= 0.89 and service = other| http | remote_job | namenetbios_ns | eco_i | mtp: then **normal**
- **Rule2:** If protocol_type = tcp|UDP|ICMP and source_bytes <= 8 and wrong_fragment > 0 then attack is **DOS**(source_bytes <= 8 and wrong_fragment > 0 and protocol_type = tcp then attack is teardrop (0.0)source_bytes <= 8 and wrong_fragment > 0 and protocol_type = udp: then attack is teardrop (892.0) source_bytes <= 8 and wrong_fragment > 0 and protocol_type = icmp: pod (198.0))
- **RULE 3:**If protocol_type = tcp and source_bytes <= 8 and count <= 2 and Destination_host_srv_diff_host_rate <= 0.48 and Destination_host_diff_srv_rate <= 0.1 and Destination_host_serror_rate <= 0.89 and service = ftp_data and Destination_host_rerror_rate <= 0.04 and Destination_host_same_source_port_rate <= 0.51 then **normal** (28.0)
- **RULE 4:** If count <= 2 and Destination_host_srv_diff_host_rate <= 0.48 and protocol_type = udp And Destination_host_diff_srv_rate <= 0.01: then attack is **U2R** rootkit (2.0)
- **RULE 5:** If protocol_type = tcp and source_bytes <= 8 and count <= 2 and Destination_host_srv_diff_host_rate <= 0.48 and Destination_host_diff_srv_rate <= 0.1 and Destination_host_serror_rate <= 0.89 and service = ftp_data and Destination_host_rerror_rate <= 0.04 and Destination_host_same_source_port_rate > 0.51 and logged_in <= 0 and Destination_bytes <= 1050583 and Destination_bytes <= 235404 then attack is warezmaster (3.0/1.0) and Destination_bytes > 235404 then attack is multihop (2.0) and Destination_bytes > 1050583 then attack is **warezmaster** (15.0) **U2R**

- **RULE 6 :**If protocol_type = tcp and source_bytes <= 8 and count <= 2 and Destination_host_srv_diff_host_rate <= 0.48 and Destination_host_diff_srv_rate <= 0.1 and Destination_host_serror_rate <= 0.89 and service = gopher| echo| discard | nntp |imap4| ssh| daytime| pop_2: (1.0) **probe**
- **RULE 7 :** If source_bytes > 8 and wrong_fragment <= 0 and source_bytes <= 16787 and Destination_host_srv_diff_host_rate <= 0.1 and Destination_bytes <= 0 service = ecr_i and source_bytes > 292: smurf (2646.0) **DOS**
- **RULE 8 :** If source_bytes > 8 and wrong_fragment <= 0 and source_bytes <= 16787 and Destination_host_srv_diff_host_rate <= 0.1 and Destination_bytes <= 0 and service = tim_i and Destination_host_diff_srv_rate <= 0.01: pod (4.0/1.0)**DOS**
- **RULE 9 :** If source_bytes <= 8 and count <= 2 and Destination_host_srv_diff_host_rate <= 0.48 and protocol_type = tcp and Destination_host_diff_srv_rate <= 0.1 and Destination_host_serror_rate <= 0.89
  and service = imap4: imap (2.0) R2L
- **RULE 10 : If** source_bytes > 8 and wrong_fragment <= 0 and source_bytes <= 16787 and Destination_host_srv_diff_host_rate <= 0.1 and Destination_bytes > 0 and hot > 0 and hot <= 25 and source_bytes <= 1551 and source_bytes <= 130 and Destination_host_diff_srv_rate <= 0.01 then R2L(GUESSPASSWORD)

5.7. Chapter summary

In this chapter, a supervised framework for intrusion detection is developed and efficiency of framework is tested. To develop framework various experiments are performed.

- For experimentation 3 basic classifiers are used.

- ❖ Decision tree classifier
- ❖ Rule based classifier
- ❖ Bayes net classifier

- All the three types of classifiers are tested with two validation strategies
  - ❖ Percentage split
  - ❖ 10 fold cross validation.
- Two data preprocessing filters are used.
  - ❖ Supervised attribute selection
  - ❖ Discritization
- Two ensemble methods are used
  - ❖ Bagging
  - ❖ boosting
- Theoretical background of classifiers, validation methods, data preprocessing and ensemble methods are described in chapter 3 whereas details experiments are given in this chapter.
- The dataset used for experimentation is NSL KDD dataset. All the experiments are evaluated on performance measurement terms like correctly classified instances, true positive rate, false positive rate and relative absolute error.
- Decision tree J48 algorithm with appropriate parameter setting when used with supervised attribute selection gives best performance amongst all experiments. Therefore, J48 algorithm is used to train model. This model generates decision tree which is provided in Appendix B. Classification of network data is completed through generated decision tree. This decision tree generates rules for classification.
- Further, this model is tested, using test data set and it gives prediction accuracy of 0.99. Therefore, the methodology adopted in this chapter to evaluate the developed framework provides good estimation of the performance.

So this chapter elaborated construction of framework for intrusion detection using supervised data mining techniques.

5.8. Chapter References

1. Attribute Relationship File Format ,http://www.cs.waikato.ac.nz/ml/ weka/arff.html.

2. The KDD Archive. KDD99 cup dataset, 1999. http://kdd.ics.uci.edu/databases/ kddcup99/kddcup99.html

3. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I.H. Witten,(2009), "The WEKA Data Mining Software: An Update" , ACM SIGKDD Explorations Newsletter, Volume 11 , Issue 1, pp. 10-18.

4. Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani ,(2009), "A Detailed Analysis of the KDD CUP 99 Data Set" ,CISDA.

Chapter 6

Observation and Findings

6.1. Introduction

This chapter discuss in detail about observation and findings based on survey performed. This research work is carried out in order to find out network security related issues and to find challenges to intrusion detection system.

6.2. Observation and findings based on survey

1. Intrusion detection systems are highly required to ensure computer network security.

   93% companies agree or strongly agree that IDS intrusion detection system is must for computer network security, Most important fact observed about network security is no single solution protects system from a variety of threats. There is need of multiple layers of security. If one fails, others still stand .Network security is accomplished through hardware and software. A network security system usually consists of many components. Ideally, combined and layered approach minimizes maintenance and improves network security. In order to strengthen the security, single tool do not provide foolproof solution. Hence a firewall and antivirus must go together with Intrusion Detection Tools.

2. Anomaly Based IDS are more suitable than Signature Based IDS for intrusion detection purpose organization.

   80% companies ,agree or strongly agree that Anomaly Based IDS are more suitable for our organization than Signature Based IDS. Anomaly based intrusion detection system identify valid network activity, so it allow only

valid activity and make detection of abnormal activity in data. Anomaly detection refers to storing features of normal behaviors into knowledgebase and compares current behavior with those in knowledgebase. Anomaly detection mainly involves the creation of knowledge bases and anomaly detection. Whereas signature based system works on signatures. Signatures are patterns to known attacks or misuses of systems. Signature detection mainly searches for signature ,signatures are specific to known attacks and they are stored in signature database. It advances in the high speed of detection and low percentage of false alarm. However, it fails if signature is missing in signature database, so it cannot detect the numerous new attacks.

3. The most critical security threat to computer network security is unauthorized access.

   63% respondents identifies most critical security threat is Unauthorized access. Unauthorized access usually refers to gaining access to any computer or network without authorization. Usually such access is obtained by extending existing privileges or stealing privileges. This is most serious security threat.

4. False alarm about intrusion is the most challenging factor to monitor intrusions using IDS

   Most critical challenge for intrusion detection system as per 53% pune IT industrial units are false alarm generation. False alarm refers to two types of alerts –first is False positive (FP) and second is false negative. False positive means network traffic is normal but identified attack whereas false negative means network traffic has attack but identified normal. Both the cases causes compromise with reliability IDS. False alarm is inversely proportional to accuracy i.e. more the false alarm; less is the accuracy.

5. Accuracy of intrusion detection is most important parameter while selecting IDS (intrusion detection system) for the security management of your organization.

   77% of pune IT industrial units says Accuracy of intrusion detection is most important. Accuracy is the proportion of the total number of predictions that

were correct; accuracy is also represented through correctly classified instances. It shows the percentage of test instances that were correctly classified. The percentage of correctly classified instances is often called accuracy or sample accuracy.

6. Security attacks are viable on any computer connected through network

   87% companies agree or strongly agree that there is strong possibility of security attack to computer. IT industries consider that intrusion attacks are viable on computers.Any computer connected through network has possibility of intrusion attack. An intrusion attack is realization of threat, the harmful action aiming to find and exploit the system vulnerability. Computer attacks causes various affect to computer ; attack destroy or access unauthorized data, may involve destroying or accessing data, threaten the computer by degrading its performance. Computer and network attacks have evolved greatly over the last few decades. The attacks are increasing in number and also improving in their strength and sophistication. The detection of intrusions in network traffic is a challenging task. In order to deal with inherent challenges, such as the ever changing environment and increasing levels of threats, there is a need for different perspectives and alternative approaches to secure systems.

7. Conf identical data is stored on the computers of IT industrial units.

   59% companies agree or strongly agree that highly confidential data is stored on the computers. Security is mandatory because confidential data is stored on the computers.

8. Network security is associated with cost (hardware cost, software cost, maintenance cost, cost of data loss, cost of incorrect decision making). Compromise with security is associated with cost.

   100% companies agree or strongly agree that Computer network security is very essential because Compromise with security affects cost. Compromise with security has financial consequences. Network security is associated with

cost (hardware cost, software cost, maintenance cost, cost of data loss, cost of incorrect decision making).

9. Antivirus and firewall together do not provide full proof solution to network security.

   64% companies consider that Antivirus and firewall together do not provide full proof solution to network security.

   Network Security Management process mainly involves components like antivirus, firewall and intrusion detection system. Antivirus, is one of most important factor of computer network security. Anti-virus prevents and gets rid of viruses. A virus programme presents harmful software from installing and damaging computer. Antivirus software protects the computer from infected files. Antivirus detects the infections in the system and heals it, depending on the updated version. Other important factor of computer network security is firewall. Firewalls act as a barrier between corporate (internal) networks and the outside world (Internet), and filter incoming traffic according to a security policy. Firewalls are not completely foolproof. A firewall generally makes pass-deny decision on the basis of allowable network addresses.

   Intrusion detection is a passive approach to security as it monitors information systems and raises alarms when security violations are founded. Examples for security violations contain the abuse of privileges or the use of attacks to exploit software or protocol vulnerabilities. The detection of intrusions in network traffic flows and host activities is a challenging task. In order to deal with inherent challenges, such as the ever changing environment and increasing levels of threats, we clearly need different perspectives and alternative approaches to secure our systems - the approaches that can adapt to drifting concepts and provide flexibility when the systems are targeted

.

6.3. Observation and findings based on experiments

1. Data mining provides useful alternative for anomaly based intusion detection
2. Decision tree based methods perform better than baysian method and rule based methods when used for intrusion detection .accuracy of J48 method gives high accuracy.
3. Information gain based feature selection methods are suitable for data preprocessing before intrusion detection.
4. Ensemble methods give slow performance for intrusion detection.
5. Supervised algorithm provides accurate predictions about intrusion attack.

6.4. Chapter summary

From the overall observation and findings it can be said that computer network security is very essential because highly confidential data is stored on computers and compromise with security causes financial consequences. Intrusion detection systems are very indispensable for computer network security. For intrusion detection usability depends upon the accuracy of detection. So there is need to develop intrusion detection framework which provides higher accuracy.

Chapter 7

Conclusions, Suggestions and Scope for future research

7.1. Introduction

This chapter gives an idea about whole research work. This research work is carried out in order to develop data mining framework for detection of intrusion attack on computer network security.

7.2. Conclusions

1. Intrusion based security attacks are challenging for Pune IT industrial units.
2. Unauthorized access of computer network is most severe security threat to Pune IT industrial units.
3. Network administrators of Pune IT industrial units, considers ‘accuracy of intrusion detection’ as most important parameter for selection of IDS.
4. Pune IT industrial units face problem of false alarm generation with existing intrusion detection system.
5. In this study, attempts have been made to use Data Mining techniques with the aim of detecting intrusion based security attacks in the computer network. Decision tree classification technique used by SIDDM framework is capable and usable for intrusion detection. This technique with appropriate parameter and feature reduction is able to better classify network activity and recognize whether it is valid or not.

6. This study proposes the supervised data mining approach SIDDM for detection of intrusion based security attack on the computer system. Supervised models are constructed from large storage of network data and once model is built it can be used to predict attack in unknown network data. It proposes method to identify threats which may serious harm to computer.

7. The proposed model has prognostic capability, for unknown network data it significantly identifies attacks. Model will offer the advantage of considering those unlabeled records. This model is used to classify the network data samples as anomalous behaviour data or the normal behaviour data. Thereby the proposed model can be greatly deployed for intrusion detection in IT industrial units.

8. This research proposes anomaly based intrusion detection through SIDDM framework. This systematic framework is developed to detect intrusion based security attack using data mininig whereas most of commercial IDS do this by statistical analysis. In this empirical work, experiments are performed using supervised classifiers on benchmarked KDD network data collection.

9. From empirical results of this research experiments, it is concluded that decision tree-J48 classifier is best and stable classifier for organizations concerned with overall correct classification of intrusion detection. The experimental results on KDD benchmark dataset evident that proposed algorithm achieved high detection rate on different types of network attacks. Comparison of all the performed experiments result shows, that the classifier with 10-fold cross validation using the J48 decision tree algorithm with the appropriate parameter values showed the best classification accuracy. This classifier model has a prediction accuracy of 99.742% on the training datasets. If J48 is used with ensemble method it takes long time for detection therefore without ensemble methods are faster and accurate for intrusion based security attack detection.

10. In summary, the results from this study contribute towards improving the networking security and give solution for detection of intrusion based security attack.

➢ Proposed framework for intrusion detection

SIDDM (Systematic Intrusion Detection using Data Mining) model is developed in this research work. Computer network security always demands, superior methods for intrusion detection. IDS are to detect all intrusions at first effectively.

This framework proposes

- To construct intrusion detection system based on supervised data mining model and step to built model are as follows.

  Construct training model using available labeled network history data, apply data preprocessing and remove unnecessary attributes from data set. On preprocessed dataset, apply supervised attribute selection to select only most relevant feature; this list is given in chapter 5 in section 5.6.1. , On reduced dataset apply 10 fold cross validation, training of model using J48 decision tree classifier. Once model is trained test using unclassified and unlabeled network data; now model is ready for intrusion detection in for new data.

- Alternatively rules constructed by this model can be incorporated in existing intrusion detection script or rule engine. these rules are given in annexure 2.

Features of framework

- SIDDM framework offers anomaly based intrusion detection for identification and categorization attacks.
- SIDDM framework uses supervised data mining method for construction of model.

Steps to use framework

Intrusion detection system using the concept of SIDDM framework uses following steps to identify whether the network data is normal or attack.

1. Collect network data using packet sniffer software.

2. Apply data preprocessing .

3. Classification the network data based on rules generated by model.

4. Attack with attack type identification will be done by model.

Advantages of SIDDM Model

- SIDDM Model strengthens computer network security by providing data mining based framework to find abnormality in data.

- SIDDM Model provides a method to construct intrusion detection system. Method proposed is useful for construction of highly accurate intrusion detection system.

- SIDDM Model detects intrusion with high accuracy. This model generates less amount of false positive and false negative.

- Intrusion attacks are classified with high efficiency i.e time taken by model construction is also less.

- Intrusion detection and Classification rules are generated. These rules are available in annexure 2.

- Rules generated by SIDDM for detection and classification of intrusion attack can be incorporated into tools like Snort (commercial intrusion detection tool), firewalls, or detection scripts to identify intrusion attack.

This thesis research contributes to both the network security and the data mining field. Below is the summary of contributions:

- This thesis presents a systematic analysis of several steps involved in a data mining process, providing both theoretical and realistic contributions.

  Data mining techniques are used to specify the kind of patterns to be found in data mining tasks. Various data mining techniques are surveyed using experiments. before construction of predictive data mining model ,supervised techniques like decision tree, rule based classification and bayes net are surveyed for their applicability in intrusion detection Predictive: to perform inference on data and to make predictions. Prediction model discovers the

relationship between dependent and independent variables. Data mining showing how particular attributes within the data will behave in future

- The Network Intrusion predictive model, which is developed in this study, generated various patterns and rules.
- Thesis determines characteristic analogy for intrusion based security attacks.
- Thesis categorizes the types of losses due to detected attacks.
- Presents, a survey of the various data mining techniques that have been proposed towards the enhancement of IDSs.
- Demonstrates, the effectiveness of supervised classification techniques in detecting anomalies.
- Analyses, components of computer network security.
- Elaborates, Intrusion Detection System, which is one of most important component of computer network security.

Intrusion Detection Systems (IDS) are the second layer of defense. It detects the presence of attacks within traffic that flows in through the holes punched into the firewall. Intrusion detection is the process of which monitors the events occurring in a computer system or network to analyze them for signs of intrusion. An Intrusion Detection System (IDS) constantly monitors actions in a certain environment and decides whether they are part of a possible hostile attack or a valid use of the environment. Following are the types of intrusion detection system.

The conclusion, of this study has shown that the data mining methods generate interesting rules that are crucial for intrusion detection and prevention in the networking industry.

This thesis attempts to address the problem of intrusion attack detection with the use of data mining supervised model. In summary, the results from this study can contribute towards in improving the networking security.

7.3. Suggestions

For computer network security in IT industries, it is imperative that IT industries must enhance security to detect intrusion based security attack. Following are the suggestions for obtaining effective complete network security.

1. Implement second layer of security compulsorily.

   Pune IT industries must apply second layer of security. First layer of security antivirus and firewall are not sufficient. Compromise with security causes serious consequences. As antivirus and firewall do not offer completely secure network therefore along with firewall and antivirus other security components are must. Therefore second layer of security needed to implement.

2. Install Anomaly based intrusion detection system as second layer of security. data mining based anomaly detection

3. Implement SIDDM Model (Data mining framework) developed in this thesis for accurate intrusion detection

   Data mining framework suggested in this research gives higher accuracy of intrusion detection. This reduces problem of false alarm and gives accurate intrusion detection. This framework gives efficient alternative way for intrusion detection.

7.4. Future scope of work

These experiments and their results provide reliable guidelines for future research in applying supervised classifiers for field of intrusion detection and expose some new avenues of research .Many improvements can be added to the intrusion detection system developed in this thesis.

The study of intrusion detection systems is quite new relative to many other areas of systems research and it stands to reason that this topic offers a number of opportunities for future exploration. There are a variety of research directions that can be further developed using part of this thesis.

- Developing, An Intrusion detection tool (Software) , using method proposed through this research.

- Investigating, applicability of unsupervised data mining techniques for intrusion detection.

- Developing, a system that operates with a more global scope may be capable of detecting distributed attacks or those that affect an entire enclave. Development of such a system would be a valuable contribution to the study of intrusion detection.

- This study was carried out using supervised data mining techniques. Supervised Classification algorithms such as J48 decision tree, rule based One R and bayes net algorithms. So further investigation needs to be done using other classification algorithms such as Neural Networks and Support Vector Machine plus using association rule discovery.

Bibliography

- Adetunmbi A.Olusola., Adeola S.Oladele. and Daramola O.Abosede ,(2010), “Analysis of KDD 99 Intrusion Detection Dataset for Selection of Relevance Features” , Proceedings of the World Congress on Engineering and Computer Science 2010 Vol I WCECS 2010, San Francisco, USA.

- Adeyinka, O.,(2008), “Internet Attack Methods and Internet Security Technology Modeling & Simulation” , AICMS 08. Second Asia International Conference on,vol., no., pp.77-82.

- Ajibuwa F. O. ,(2006), “ Data and Information Security in Modern Day Businesses” ,Published M.Sc. dissertation, Atlantic International University, U.S. ,from http://www.aiu.edu/publications/student/ english/Data and Information Security in Modern Day Businesses thesis.html .

- An Introduction to Computer Security: The NIST Handbook ,Special Publication 800-12.

- Antivirus - how-antivirus-software-works , http://www.howtogeek.com /125650/ htg-explains-how-antivirus-software-works.

- Attribute Relationship File Format ,http://www.cs.waikato.ac.nz/ml/ weka/arff.html.

- Basta A., Halton W.,(2003) “Computer Security- Concepts, Issues and Implementation”, New Delhi: Course technology/cengage Learning.

- Bishwanath mukharjee L.Todd heberlien,(1994), “Network Intrusion Detection” ,IEEE.

- Bhavya Daya ,(2010), “Network Security: History, Importance, and Future”, University of Florida Department of Electrical and Computer Engineering.

- Bro network security monitor , www.bro.org, Accessed date Jan 2012

- C. F. Tsai, Y. F. Hsu, C. Y. Lin and W. Y. Lin, (2009), " Intrusion detection by machine learning: A review" ,, Expert Systems with Applications, Vol 36, Issue 10, pp. 11994-12000. 2009.

- Carl F.,(2003), "Intrusion Detection and Prevention", McGraw-Hill, Osborne Media.

- Chia-Mei Chen , Ya-Lin Chen, Hsiao-Chung Lin,(2010) "An efficient network intrusion detection" ,Computer Communications 33 ,477–484.

- Cisco intrusion detection , www.cisco.com, Accessed date Jan 2012

- Counteract edge for threat prevention www.forescout.com/product /counteract-edge, Accessed date Jan 2012

- Curtin M., (1997),"Introduction to Network Security," http://www.interhack.net/ pubs/network-security.

- Daniel Barbara, Julia C., (2001), "ADAM: Detecting Intrusions by Data Mining" , Proceedings of the 2001 IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY, 5.

- Dash M. and Liu H.,(1997), "Feature selection for classification", Intelligent Data Analysis: An International Journal, PP. 131–156.

- Dewan Md. Farid, Nouria Harbi, Emna Bahri, Mohammad Zahidur Rahman, Chowdhury Mofizur Rahman ,(2010), "Attacks Classification in Adaptive Intrusion Detection using Decision Tree", World Academy of Science, Engineering and Technology 63.

- Dorothy E. Denning , (1987),"An Intrusion-Detection Model", IEEE Transactions On Software Engineering, Vol. Se-13, No. 2, , 222-232. .

- Dr.S.Siva Sathya , Dr. R.Geetha Ramani , K.Sivaselvi ,(2011), "Discriminant Analysis based Feature Selection in KDD Intrusion Dataset" ,International Journal of Computer Applications (0975 – 8887) Volume 31– No.11.

- Dragon documentation http://www.nuance.com/naturallyspeaking/resources/ documents/ usergd_v11.pdf , Accessed date Jan 2012

- E. Hernandez-Pereira, J. A. Suarez-Romero, O. Fontenla-RomeroA. Alonso-Betanzos, ,(2009), " Conversion methods for symbolic features: A comparison applied to an intrusion detection problem" , Expert Systems with Applications.

- E.Kesavulu Reddy, Member IAENG, V.Naveen Reddy, P.Govinda Rajulu,(2011), " A Study of Intrusion Detection in Data Mining " ,Proceedings of the World Congress on Engineering 2011 Vol III WCE 2011, , London, U.K. ISSN: 2078-0966 (Online).

- Ellen Pitt and Richi Nayak, (2007),"The Use of Various Data Mining and Feature Selection Methods in the Analysis of a Population Survey Dataset",Conferences in Research and Practice in Information Technology.

- Enterprise network security- airmagnet |flukenetworks www.flukenetworks.com , Accessed date Jan 2012

- Everything you need to know about network security, www.axent.com , last accessed 2012.

- Fangfei Weng, Qingshan Jiang, Liang Shi, and Nannan Wu,(2007), "An Intrusion Detection System Based on the Clustering Ensemble", IEEE.

- Flora S. Tsai ,(2009), "Network Intrusion Detection using Association Rules" International Journal of Recent Trends in Engineering, Vol 2, No. 2.

- G. Kumar, K. Kumar and M. Sachdeva, " The Use of Artificial Intelligence based Techniques For Intrusion Detection – A Review, Artificial Intelligence Review" , vol. 34, No. 4, pp. 369-387, Springer, Netherlands, DOI: 10.1007/s10462-010-9179-5 ISSN: 0269-2821.

- G. Kumar, K. Kumar, M. Sachdeva,(2010), " An Empirical Comparative Analysis of Feature Reduction Methods For Intrusion Detection" , International Journal of Information and Telecommunication, 1 , 44-51, ISSN: 0976-5972.

- G.V. Nadiammai, M. Hemalatha ,(2014),"Effective approach toward Intrusion Detection System using data mining techniques", Cairo University,Egyptian Informatics Journal,www.elsevier.com/locate/eij ,1110-8665 .

- Geoff Norman, (2010), " Likert scales, levels of measurement and the laws of statistics", Springer Science Business Media B.V.

- Ghanshyam Prasad Dubey, Prof. Neetesh Gupta, Rakesh K Bhujade,(2011), " A Novel Approach to Intrusion Detection System using Rough Set Theory and Incremental SVM" ,International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-1.

- Gregory Piatetsky-Shapiro, Christopher Matheus, Padhraic, Smyth, and R amasamy Uthurusamy,(1994), " KDD-93: Progress and Challenges in Knowledge Discovery in Databases"

- Gunja Ambica et al.(2012) ," Robust Data Clustering Algorithms for Network Intrusion Detection" , International Journal of Computer & Organization Trends –Volume2Issue5- 2012 ISSN: 2249-2593 Page 118

- H. Gunes Kayacık, A. Nur Zincir-Heywood, Malcolm I. Heywood,. " Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99Intrusion Detection Datasets", Dalhousie University, Faculty of Computer Science, http://www.cs. dal. ca /projectx/ ,2006 .

- Harry N Boone , Deborah A Boone, (2012),"analyzing likert data", journal of extension, vol 50.

- Heberlein L., Dias G., Levitt K., Mukherjiee B., Wood J., and Wolber D.,(1990), "A Network Security Monitor" , Proc., IEEE Symposium on Research in Security and Privacy, Oakland, CA, pp.196-304.

- Hershkop S., Apap F., Eli G., Tania D., Eskin E., Stolfo S., (2007),"A data mining approach to host based intrusion detection" , Technical reports, CUCS Technical Report.

- Hulus onder, (2007),“A security management system design” , thesis , middle east technical university.

- Huu Hoa Nguyen, Nouria Harbi and Jerome Darmont, (2011)“An Efficient Fuzzy Clustering-Based Approach for Intrusion Detection”

- I. Elaine Allen and Christopher A Seaman ,(2007),” likert scales and data analyses”, statistics round table.

- Ilia Mitov, Krassimira Ivanova, Krassimir Markov,Vitalii Velychko, Peter Stanchev, Koen Vanhoof, (2008), “comparison of discretization methods for preprocessing data for pyramidal growing network classification method” , International Book Series "Information Science and Computing".

- IT companies in and around pune , www.punediary.com

- Jiawei Han And Micheline Kamber,(2008), “Data mining concepts and techniques” , Morgan Kaufmann publishers .an imprint of Elsevier .ISBN 978-1-55860-901-3. Indian reprint ISBN 978-81-312- 0535-8 .

- John Mchugh, (2001),"Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory" .

- John McHugh, Alan Christie, and Julia Allen ,(2000), “The Role of Intrusion Detection Systems”, IEEE SOFTWARE .

- John Wack, Ken Cutler, Jamie Pole,(2002), “Guidelines on Firewalls and Firewall Policy ” ,Recommendations of the National Institute of Standards and Technology.

- Johnson R.,(2002), “Applied Multivariate Statistical Analysis”, Prentice Hall, PP. 356-395.

- Joyce Jackson, (2002), “Data Mining: A Conceptual Overview” Communications of the Association for Information Systems ,Volume 8.

- Juniper network intrusion detection www.juniper.net, Accessed date Jan 2012

- Kamran Shafi, (2008),"An Online and Adaptive Signature-based Approach for Intrusion Detection Using Learning Classifier Systems" ,THESIS, University of New South Wales.

- Karamjeet kaur , "a survey of intrusion detection techniques" International Journal of Advanced Research in Computer Science and Software Engineering,2013.

- Karen S. and Peter M.,(2007), "Guide to Intrusion Detection and Prevention Systems", National Institute of Standards and Technology, Department of Commerce, USA.

- Kayacik, G. H., Zincir-Heywood, A. N.,(2005), "Analysis of Three Intrusion Detection System Benchmark Datasets Using Machine Learning Algorithms" , Proceedings of the IEEE ISI 2005 Atlanta, USA.

- KdNuggets,(2007), "Data Mining Methodology", http://www.kdnuggets.com /polls/2007/ datamining_methodology.htm,

- Kendall, K.,(1999) " A database of computer attacks for the evaluation of intrusion detection systems" , Masters thesis, Massachusetts Institute of Technology.

- Kevin J. Houle,(2001), "Trends in Denial of ServiceAttack Technology", CERT Coordination Center.

- Kingsly Leung Christopher Leckie y.,(2005), "Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters" ,28th Australasian Computer Science Conference, The University of Newcastle, Australia,. Conferences in Research and Practice in Information Technology, Vol. 38.

- Kiran Dhangar Prof. Deepak Kulhare Arif Khan, "Intrusion Detection System (A Layered Based Approach for Finding Attacks" , International Journal of Advanced Research in Computer Science and Software Engineering,may 2013

- Kothari C. R. ,(2004), "Research Methodology, Methods and techniques" (2nd ed.), New Delhi: New age International (p) Ltd.

- L. Zenghui, L. Yingxu, “A Data Mining Framework for Building Intrusion Detection Models Based on IPv6” ,Proceedings of the 3rd International Conference and Workshops on Advances in Information Security and Assurance. Seoul, Korea, Springer- Verlag, 2009.

- Laura Ruotsalainen,(2008), Data Mining Tools for Technology and Competitive Intelligence,VIT, ISBN 978-951-38-7240-3.

- Levent Ertoz, Eric Eilertson, Aleksandar Lazarevicy, Pang-Ning Tan, Vipin Kumar, Jaideep Srivastava_y, Paul Dokas., (2004), ”The MINDS – Minnesota Intrusion Detection System” .

- M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I.H. Witten,(2009), “The WEKA Data Mining Software: An Update” , ACM SIGKDD Explorations Newsletter, Volume 11 , Issue 1, pp. 10-18.

- M. Xue, C. Zhu, “Applied Research on Data Mining Algorithm in Network Intrusion Detection” ,jcai, pp.275-277, 2009 International Joint Conference on Artificial Intelligence, 2009.

- M.Govindarajan and RM.Chandrasekaran, “Intrusion Detection using an Ensemble of Classification Methods” ,Proceedings of the World Congress on Engineering and Computer Science 2012 Vol I WCECS 2012, , San Francisco, USA, 2012.

- M.Vijayakamal, Mulugu Narendhar,(2012), “A Novel Approach for WEKA & Study On Data Mining Tools”, International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 2.

- Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani ,(2009), “A Detailed Analysis of the KDD CUP 99 Data Set” ,CISDA.

- Marin, G.A.,(2005), "Network security basics Security & Privacy” ,, IEEE , vol.3, no.6, pp. 68-72.

- Mcafee network intrusion prevention, www.mcafee.com ,Accessed date Jan 2012

- Meenakshi.RM, Mr.E.Saravanan,(2013), “A data mining analysis & approach with intrusiondetection / prevention from real”.

- Michael J. Pazzani , (2000), “Knowledge discovery from DATA?”, IEEE Intelligent Systems.

- Michael Wilkison,(2005), “ Intrusion Detection FAQ: How to Evaluate Network Intrusion Detection Systems?” http://www.sans.org/security-resources/idfaq/eval_ids.php

- Mithcell Rowton,(2005), “Introduction to Network Security Intrusion Detection” , December 2005.

- Mostaque Md. Morshedur Hassan,(2013), current studies on intrusion detection system, genetic algorithm and fuzzy logic International Journal of Distributed and Parallel Systems (IJDPS) Vol.4, No.2.s

- Mohammadreza Ektefa , Sara Memar, Fatimah Sidi, Lilly Suriani Affendey.,(2010),“Intrusion Detection Using Data Mining Techniques ” , 978-1-4244-5651-2/10/2010.

- Muamer N. Mohammad, Norrozila Sulaiman, Osama Abdulkarim Muhsin, (2011),“A Novel Intrusion Detection System by using Intelligent Data Mining in Weka Environment”, sciencedirect Procedia Computer Science 3 1237–1242 .

- Nagaraju Devarakonda, Srinivasulu Pamidi, Valli Kumari V, Govardhan, “ Intrusion Detection System using Bayesian Network and Hidden Markov Model”, elsevier,2012.

- Natesan, P.,P. Balasubramanie and G. Gowrison “Improving the Attack Detection Rate in Network Intrusion Detection using Adaboost Algorithm”, Journal of Computer Science 8 (7): 1041-1048, 2012 ISSN 1549-3636, Science Publications,2012.

- Nguyen H.A., Choi D. , “Application of Data Mining to Network Intrusion Detection: Classifier Selection Model” , APNOMS, LNCS 5297, pp. 399–408, 2008.

- Nigel Williams, Sebastian Zander, Grenville Armitage, “A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification”, ACM SIGCOMM Computer Communication Review Volume 36, Number 5, r 2006 .

- Oyebode E.O., Fashoto S.G.,Ojesanmi O.A. and Makinde O.E. “Intrusion Detection System for Computer Network Security”, Australian Journal of Basic and Applied Sciences, 5(12): 1317-1320, ISSN 1991-8178,2011.

- Ozgur Depren, Murat Topallar, Emin Anarim, M. Kemal Ciliz. “An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks” , Expert Systems with Applications 29,713–722,www.elsevier.com/locate/eswa,2005.

- Nagaraju Devarakonda et al. “Intrusion Detection System using Bayesian Network and Hidden Markov Model”, Available online at www.sciencedirect.com, elsevier , Procedia Technology 4 (2012) 506 – 514

- P. Garcıa-Teodoroa, J. Dıaz-Verdejo, G. Macia-Fernandez, E. Vazquez,(2009), “Anomaly-based network intrusion detection:Techniques, systems and challenges” ,Sciencedirect, 2009.

- P. K. Srimani and Manjula Sanjay Koti,(2012), “Medical Diagnosis Using Ensemble Classifiers - A Novel Machine-Learning Approach”.

- Panos Louvieris, Natalie Clewley, Xiaohui Liu . “Effects-Based Feature Identification for Network Intrusion Detection” ,2013.

- Pavel Laskov, Patrick Dussel, Christin Schafer and Konrad Rieck, “Learning intrusion detection: supervised or unsupervised?” ,2001.

- Pune IT software companies in pune list, www.pune.ws

- Quinlan, J.R. ,(1993), “C4.5: Programs For Machine Learning”, San Mateo, CA: Morgan Kaufmann.

- Rachid Beghdad,(2009), “Efficient deterministic method for detecting new U2R attacks”, Computer Communications 32 , 1104–1110.

- Rajashree Dash, Rajib Lochan Paramguru, Rasmita Dash,(2011) “Comparative Analysis of Supervised and Unsupervised Discretization Techniques”.

- Rajni jain ,(2011), “Introduction To Datamining Techniques” ,www.iasri.res.in/ebook/ expertsystem/DataMining.pdfSimilar

- Ramageri B. M. ,(2011), “Data Mining Techniques and Applications” , In Indian Journal of Computer Science and Engineering Vol. 1 No. 4 pp 301-305,2011.

- Richard Kirkby, (2011),Eibe Frank ,WEKA Explorer User Guide.

- Rodriguez, J.J. and L.I. Kuncheva,(2009), “Rotation Forest: A New Classifier Ensemble Method” , IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(10): 1619-1630s.

- Roman V. Yampolskiy and Venu Govindaraju, (2007),“Computer Security: a Survey of Methods and Systems “, Journal of Computer Science 3 (7): 478-486.

- Ronaldo C. Prati, Gustavo E.A.P.A. Batista, and Maria Carolina Monard,(2011), “A Survey on Graphical Methods for Classification Predictive Performance Evaluation” , IEEE Transactions on Knowledge And Data Engineering, VOL. 23, NO. 11.

- S.S.Joshi (2010). “A Study of Information Security Policies in Selected IT Companies in Pune City”. Published Doctorial dissertation, University of Pune. Retrieved March 3, 2011 from http://shodhganga.inflibnet.ac.in/handle /10603/2026.

- S.Vijayasankari, K. Ramar , (2012), “Enhancing Classifier Performance Via Hybrid Feature Selection and Numeric Class Handling- A Comparative Study”, International Journal of Computer Applications (0975 – 8887) Volume 41– No.17.

- Sahilpreet Singh Meenakshi Bansal ,(2013), Improvement of Intrusion Detection System in Data Mining using Neural Network ,Volume 3, Issue 9, September 2013 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering

- Saumil Shah, “the Anti Virus Book”, The Tata McGraw-Hill Publishing Company Ltd. http://saumil.net/antivirus

- Serhat Ozekes, Onur Osman,(2013), “Classification And Prediction In Data Mining With Neural Networks” ,Journal Of Electrical & Electronics Engineering.

- Shima Aghtar ,“A New Incremental Classification Approach Monitoring The Risk of Heart Disease” ,THESIS McMaster University DigitalCommons @McMaster, 2012

- Sherish Johri ,(2012), “Novel Method for Intrusion Detection using Data Mining “Volume 2, Issue 4, April 2012 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering.

- Shyara Taruna R., (2013),“Enhanced Naïve Bayes Algorithm for Intrusion Detection in Data Mining” , (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (6) , 2013, 960-962

- Snort user manual. http://www.snort.org/docs, Accessed date Jan 2012

- Sourcefire |network security solutions www.sourcefire.com

- Stallings W. & Brown L., (2008),“Computer Security Principles and Practice”, South Asia: Pearson Education, Inc. (ISBN: 978-81-317-3351-6).

- Strata guard, www.securitywizardry.com Accessed date Jan 2012
- Sunita Beniwal , Jitender Arora,(2012), “Classification and Feature Selection Techniques in Data Mining” , International Journal of Engineering Research & Technology (IJERT),2012
- T. Bhavani et al., “Data Mining for Security Applications” , Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing - Volume 02, IEEE Computer Society,. ACM, 2008.
- T. Lappas and K. P. ,(2007), “Data Mining Techniques for (Network) Intrusion Detection System” , 2007.
- Tadeusz Pietraszek, Axel Tanner, “Data mining and machine learning—Towards reducing false positives in intrusion detection” ,2005.
- Thair Nu Phyu,(2009), “Survey of Classification Techniques in Data Mining” ,Proceedings of the International MultiConference of Engineers and Computer Scientists Hong Kong ,2009 .
- The KDD Archive. KDD99 cup dataset, 1999. http://kdd.ics.uci.edu/databases/ kddcup99/kddcup99.html
- Tim lane, f(2007), “ Information security management in australian universities:an exploratory analysis” ,thesis, qft faculty of information technology.
- Tom Mitchell , (1997),“Machine Learning”, McGraw Hill.
- Two Crows Corporation, (2005)“Introduction to Data Mining and Knowledge Discover”, 3rd edition, MD 20854(USA).
- U. Fayyad, D. Haussler, and P. Stolorz. ,(1996), “From Data Mining to Knowledge Discovery in Databases” , 0738-4602-1996 ,1996.
- U.Fayyad, G.Piatetsky, and P.Smyth, (1996),“ The KDD process for Extracting Useful Knowledge from Volumes of Data”, Communications of the ACM, Vol. 39, PP. 27-34, 1996.

- UCI Machine Learning Data Repository" , http://archive.ics.uci.edu/ml ,last accessed on Jan, 2012.
- Umesh Kumar Pandey S. Pal,(2011), "Data Mining : A prediction of performer or underperformer using classification", International Journal of Computer Science and Information Technologies.
- Venkatadri.M, Dr. Lokanatha C. Reddy, (2011), " A Review on Data mining from Past to the Future", International Journal of Computer Applications.
- Weka – ARFF file format .http://weka.wikispaces.com/ARFF bookersion
- WEKA: Waikato environment for knowledge analysis .http://www.cs.waikato.ac. nz/ml/weka C58
- Wenke Lee, (2002),"Applying Data Mining to Intrusion Detection: the Quest for Automation, Efficiency, and Credibility", SIGKDD.
- Wenke Lee,(1996), "A Data Mining Framework for Building Intrusion Detection Models",DAPRA.
- Whitman M. E. & Mattord H. J. , (2007), "Principles of Information Security" (2nd ed.), New Delhi: Thomson Learning/Course Technology.
- William J. Frawley, Gregory Piatetsky-Shapiro, and Christopher J. Matheus,(1992), "Knowledge Discovery in Databases: An Overview",AAAI,1992.
- Witten IH, Frank E. , (2005)," Data Mining: Practical Machine Learning Tools and Techniques" , Second edition, Morgan Kaufmann,2005.
- Wu Junqi1, Hu Zhengbing.,(2008), " Study of Intrusion Detection Systems (IDSs) in Network Security".,ISSN 978-1-4244-2108-4/08/, 2008 IEEE
- XindongWu ,Vipin Kumar, (2007),"Top 10 algorithms in data mining" ,survey paper, Springer-Verlag London Limited.

- Yang Q., and Wu X.,(2006), “10 Challenging Problems in Data Mining Research”, International Journal of Information Technology and Decision Making, World Scientific Publishing Company , Vol. 5, No. 4, PP.597–604.

- Zachary Miller, William Deitrick, Wei Hu, (2011), “Anomalous Network Packet Detection Using Data Stream Mining” , scientific research, Journal of Information Security.

A Survey about need and challenges of Network Security Management in IT industrial units of Pune Region

Dear Participant please note that:
This questionnaire is purely for academic research purpose. The confidentiality of the data entered by you in this questionnaire will be maintained.

Name _____Contact Number_____

Designation _____Experience in years_____

Department _____No. of IT related employees_____

Company name _____Company address_____

Kindly tick (✓) only one option which is closer to your opinion.

Network security issues		Strongly disagree	Disagree	Neither agree nor Disagree	Agree	Strongly agree
1	Intrusion based Security attacks are viable on any computer connected through network.					
2	Highly confidential data is stored on the computers of your organization.					
3	Computer security is an accountability of everyone in the organization.					
4	Computer network security is very essential because Compromise with security affects cost (hardware cost, software cost, maintenance cost, cost of data loss, cost of incorrect decision making).					
5	Having both antivirus and firewall makes your computer network completely secure.					
6	Intrusion Detection System (IDS) is must for effective network security management.					
7	Anomaly Based IDS are more suitable for our organization than Signature Based IDS.					

Kindly tick (✓) only one option which is closer to your opinion.

8. What do you see as the most critical security threat to computer network security?

A. Unauthorized access.
B. Virus/worm attack.
C. Malicious attack
D. Denial of service.

9. According to you which one is challenge to monitor intrusions using IDS ?

A. Identifying type of intrusion
B. False alarm about intrusion.
C. Alerting Mechanisms.
D. Updating Signatures/Policies.

10. Which is most important parameter while selecting intrusion detection system for the security management of your organization?

A. Product popularity
B. Capacity to detect new intrusion
C. Best user interface
D. Accuracy of intrusion detection.

Respondent signature

Result of experiments

Classification Tree constructed by Model

Framework developed in this research work uses decision tree classification technique for classification of network data. Decision tree generated in this research uses j48 decision tree with appropriate parameter setting. This tree shows features and associated values of features which decide whether it is attack data or normal. This tree is constructed using 20 attributes discussed in chapter 5.

To understand generated tree consider following branch of tree.

src_bytes <= 8
| count <= 2
| | dst_host_srv_diff_host_rate <= 0.48
| | | protocol_type = tcp
| | | | dst_host_diff_srv_rate <= 0.1
| | | | | dst_host_serror_rate <= 0.89
| | | | | | service = ftp_data
| | | | | | | dst_host_rerror_rate <= 0.04
| | | | | | | | dst_host_same_src_port_rate <= 0.51: normal

This branch of tree indicates following rule.

This tree is presetting rule to identify whether network data has attack or it is normal. Sample is as follows

If (src_bytes <= 8) and (count <= 2) and
(dst_host_srv_diff_host_rate <= 0.48) and
(protocol_type = tcp) and (dst_host_diff_srv_rate <= 0.1) and
(dst_host_serror_rate <= 0.89) and (service = ftp_data) and (dst_host_rerror_rate <= 0.04) and
(dst_host_same_src_port_rate <= 0.51)
Then network data is normal

Generated tree is shown below .

Filename: SIDDM.model

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: KDDTrain+with attacktype-weka.filters.unsupervised.attribute.Remove-R43-weka.filters.supervised.attribute.AttributeSelection-Eweka.attributeSelection.CfsSubsetEval-Sweka.attributeSelection.BestFirst -D 1 -N 5

Attributes: 20

=== Classifier model ===

J48 pruned tree

----

src_bytes <= 8

| count <= 2

| | dst_host_srv_diff_host_rate <= 0.48

| | | protocol_type = tcp

| | | | dst_host_diff_srv_rate <= 0.1

| | | | | dst_host_serror_rate <= 0.89

| | | | | | service = ftp_data

| | | | | | | dst_host_rerror_rate <= 0.04

| | | | | | | | dst_host_same_src_port_rate <= 0.51: normal (28.0)

| | | | | | | | dst_host_same_src_port_rate > 0.51

| | | | | | | | | logged_in <= 0

| | | | | | | | | | dst_bytes <= 1050583

| | | | | | | | | | | dst_bytes <= 235404: warezmaster (3.0/1.0)

| | | | | | | | | | | dst_bytes > 235404: multihop (2.0)

| | | | | | | | | | dst_bytes > 1050583: warezmaster (15.0)

| | | | | | | | | logged_in > 0

| | | | | | | | | | dst_host_srv_diff_host_rate <= 0.03: buffer_overflow (5.0/1.0)

| | | | | | | | | | dst_host_srv_diff_host_rate > 0.03: normal (5.0/1.0)

| | | | | | | dst_host_rerror_rate > 0.04

| | | | | | | dst_host_diff_srv_rate <= 0.01: ipsweep (12.0)

| | | | | | | | dst_host_diff_srv_rate > 0.01: portsweep (3.0)

| | | | | | service = other: normal (39.0)

| | | | | | service = private

| | | | | | | dst_host_rerror_rate <= 0.92: portsweep (70.0/2.0)

| | | | | | | dst_host_rerror_rate > 0.92: neptune (3.0)

| | | | | | service = http: normal (2644.0/1.0)

| | | | | | service = remote_job: normal (0.0)

| | | | | | service = name: normal (0.0)

| | | | | | service = netbios_ns: normal (0.0)

| | | | | | service = eco_i: normal (0.0)

| | | | | | service = mtp: normal (0.0)

| | | | | | service = telnet

| | | | | | | dst_host_same_src_port_rate <= 0.14: normal (36.0)

| | | | | | | dst_host_same_src_port_rate > 0.14

| | | | | | | | flag = SF: normal (0.0)

| | | | | | | | flag = S0: neptune (2.0)

| | | | | | | | flag = REJ: normal (0.0)

| | | | | | | | flag = RSTR: normal (0.0)

| | | | | | | | flag = SH: normal (0.0)

| | | | | | | | flag = RSTO: normal (2.0)

| | | | | | | | flag = S1: normal (0.0)

| | | | | | | | flag = RSTOS0: normal (0.0)

| | | | | | | | flag = S3: normal (0.0)

| | | | | | | | flag = S2: normal (0.0)

| | | | | | | | flag = OTH: normal (0.0)

| | | | | | service = finger: normal (267.0/1.0)

| | | | | | service = domain_u: normal (0.0)

| | | | | | service = supdup: normal (0.0)

| | | | | | service = uucp_path: normal (0.0)

| | | | | | service = Z39_50: normal (0.0)

| | | | | | service = smtp: normal (90.0/1.0)

| | | | | | service = csnet_ns: normal (0.0)

| | | | | | service = uucp: normal (0.0)

| | | | | | service = netbios_dgm: normal (0.0)

| | | | | | service = urp_i: normal (0.0)
| | | | | | service = auth: normal (16.0)
| | | | | | service = domain: normal (1.0)
| | | | | | service = ftp: normal (13.0/1.0)
| | | | | | service = bgp: normal (0.0)
| | | | | | service = ldap: normal (0.0)
| | | | | | service = ecr_i: normal (0.0)
| | | | | | service = gopher: satan (1.0)
| | | | | | service = vmnet: normal (0.0)
| | | | | | service = systat: portsweep (3.0)
| | | | | | service = http_443: normal (0.0)
| | | | | | service = efs: normal (0.0)
| | | | | | service = whois: normal (0.0)
| | | | | | service = imap4: imap (2.0)
| | | | | | service = iso_tsap: normal (0.0)
| | | | | | service = echo: portsweep (5.0)
| | | | | | service = klogin: normal (0.0)
| | | | | | service = link: normal (0.0)
| | | | | | service = sunrpc: normal (0.0)
| | | | | | service = login: normal (0.0)
| | | | | | service = kshell: normal (0.0)
| | | | | | service = sql_net: normal (0.0)
| | | | | | service = time: normal (76.0)
| | | | | | service = hostnames: normal (0.0)
| | | | | | service = exec: normal (0.0)
| | | | | | service = ntp_u: normal (0.0)
| | | | | | service = discard: portsweep (3.0)
| | | | | | service = nntp: satan (1.0)
| | | | | | service = courier: normal (0.0)
| | | | | | service = ctf: normal (0.0)
| | | | | | service = ssh: portsweep (1.0)
| | | | | | service = daytime: portsweep (3.0)
| | | | | | service = shell: normal (0.0)
| | | | | | service = netstat: normal (0.0)

| | | | | | service = pop_3: normal (0.0)
| | | | | | service = nnsp: normal (0.0)
| | | | | | service = IRC: normal (7.0)
| | | | | | service = pop_2: satan (1.0)
| | | | | | service = printer: normal (0.0)
| | | | | | service = tim_i: normal (0.0)
| | | | | | service = pm_dump: normal (0.0)
| | | | | | service = red_i: normal (0.0)
| | | | | | service = netbios_ssn: normal (0.0)
| | | | | | service = rje: normal (0.0)
| | | | | | service = X11: normal (2.0)
| | | | | | service = urh_i: normal (0.0)
| | | | | | service = http_8001: normal (0.0)
| | | | | | service = aol: normal (0.0)
| | | | | | service = http_2784: normal (0.0)
| | | | | | service = tftp_u: normal (0.0)
| | | | | | service = harvest: normal (0.0)
| | | | | dst_host_serror_rate > 0.89
| | | | | | dst_host_diff_srv_rate <= 0
| | | | | | | land <= 0
| | | | | | | | service = ftp_data: normal (0.0)
| | | | | | | | service = other: normal (0.0)
| | | | | | | | service = private: normal (1.0)
| | | | | | | | service = http
| | | | | | | | | dst_host_srv_diff_host_rate <= 0.01: neptune (2.0)
| | | | | | | | | dst_host_srv_diff_host_rate > 0.01: normal (23.0)
| | | | | | | | service = remote_job: normal (0.0)
| | | | | | | | service = name: normal (0.0)
| | | | | | | | service = netbios_ns: normal (0.0)
| | | | | | | | service = eco_i: normal (0.0)
| | | | | | | | service = mtp: normal (0.0)
| | | | | | | | service = telnet: normal (0.0)
| | | | | | | | service = finger: normal (0.0)
| | | | | | | | service = domain_u: normal (0.0)

| | | | | | | | **service = supdup: normal (0.0)**

| | | | | | | | **service = uucp_path: normal (0.0)**

| | | | | | | | **service = Z39_50: normal (0.0)**

| | | | | | | | **service = smtp: normal (0.0)**

| | | | | | | | **service = csnet_ns: normal (0.0)**

| | | | | | | | **service = uucp: normal (0.0)**

| | | | | | | | **service = netbios_dgm: normal (0.0)**

| | | | | | | | **service = urp_i: normal (0.0)**

| | | | | | | | **service = auth: normal (0.0)**

| | | | | | | | **service = domain: normal (0.0)**

| | | | | | | | **service = ftp: normal (0.0)**

| | | | | | | | **service = bgp: normal (0.0)**

| | | | | | | | **service = ldap: normal (0.0)**

| | | | | | | | **service = ecr_i: normal (0.0)**

| | | | | | | | **service = gopher: normal (0.0)**

| | | | | | | | **service = vmnet: normal (0.0)**

| | | | | | | | **service = systat: normal (0.0)**

| | | | | | | | **service = http_443: normal (0.0)**

| | | | | | | | **service = efs: normal (0.0)**

| | | | | | | | **service = whois: normal (0.0)**

| | | | | | | | **service = imap4: imap (2.0)**

| | | | | | | | **service = iso_tsap: normal (0.0)**

| | | | | | | | **service = echo: normal (0.0)**

| | | | | | | | **service = klogin: normal (0.0)**

| | | | | | | | **service = link: normal (0.0)**

| | | | | | | | **service = sunrpc: normal (0.0)**

| | | | | | | | **service = login: normal (0.0)**

| | | | | | | | **service = kshell: normal (0.0)**

| | | | | | | | **service = sql_net: normal (0.0)**

| | | | | | | | **service = time: normal (0.0)**

| | | | | | | | **service = hostnames: normal (0.0)**

| | | | | | | | **service = exec: normal (0.0)**

| | | | | | | | **service = ntp_u: normal (0.0)**

| | | | | | | | **service = discard: normal (0.0)**

| | | | | | | service = nntp: normal (0.0)

| | | | | | | service = courier: normal (0.0)

| | | | | | | service = ctf: normal (0.0)

| | | | | | | service = ssh: normal (0.0)

| | | | | | | service = daytime: normal (0.0)

| | | | | | | service = shell: normal (0.0)

| | | | | | | service = netstat: normal (0.0)

| | | | | | | service = pop_3: normal (0.0)

| | | | | | | service = nnsp: normal (0.0)

| | | | | | | service = IRC: normal (0.0)

| | | | | | | service = pop_2: normal (0.0)

| | | | | | | service = printer: normal (0.0)

| | | | | | | service = tim_i: normal (0.0)

| | | | | | | service = pm_dump: normal (0.0)

| | | | | | | service = red_i: normal (0.0)

| | | | | | | service = netbios_ssn: normal (0.0)

| | | | | | | service = rje: normal (0.0)

| | | | | | | service = X11: normal (0.0)

| | | | | | | service = urh_i: normal (0.0)

| | | | | | | service = http_8001: normal (0.0)

| | | | | | | service = aol: normal (0.0)

| | | | | | | service = http_2784: normal (0.0)

| | | | | | | service = tftp_u: normal (0.0)

| | | | | | | service = harvest: normal (0.0)

| | | | | | land > 0: land (8.0/2.0)

| | | | | dst_host_diff_srv_rate > 0

| | | | | | flag = SF: neptune (0.0)

| | | | | | flag = S0: neptune (135.0)

| | | | | | flag = REJ: neptune (0.0)

| | | | | | flag = RSTR: neptune (0.0)

| | | | | | flag = SH: neptune (0.0)

| | | | | | flag = RSTO: normal (2.0)

| | | | | | flag = S1: neptune (0.0)

| | | | | | flag = RSTOS0: neptune (0.0)

| | | | | | | flag = S3: neptune (0.0)

| | | | | | | flag = S2: neptune (0.0)

| | | | | | | flag = OTH: neptune (0.0)

| | | | dst_host_diff_srv_rate > 0.1

| | | | | dst_host_serror_rate <= 0.44

| | | | | | dst_host_rerror_rate <= 0.07: normal (80.0/10.0)

| | | | | | dst_host_rerror_rate > 0.07

| | | | | | | dst_host_same_src_port_rate <= 0.07

| | | | | | | | dst_host_diff_srv_rate <= 0.7

| | | | | | | | | dst_host_rerror_rate <= 0.16: satan (5.0)

| | | | | | | | | dst_host_rerror_rate > 0.16: normal (16.0)

| | | | | | | | dst_host_diff_srv_rate > 0.7: ipsweep (122.0)

| | | | | | | dst_host_same_src_port_rate > 0.07

| | | | | | | | logged_in <= 0

| | | | | | | | | dst_host_same_src_port_rate <= 0.25

| | | | | | | | | | dst_host_rerror_rate <= 0.27

| | | | | | | | | | | dst_host_diff_srv_rate <= 0.32: portsweep (178.0/3.0)

| | | | | | | | | | | dst_host_diff_srv_rate > 0.32: normal (3.0)

| | | | | | | | | | dst_host_rerror_rate > 0.27

| | | | | | | | | | | dst_host_diff_srv_rate <= 0.87: normal (12.0)

| | | | | | | | | | | dst_host_diff_srv_rate > 0.87: ipsweep (18.0)

| | | | | | | | | dst_host_same_src_port_rate > 0.25: portsweep (2188.0/8.0)

| | | | | | | | logged_in > 0: ipsweep (16.0)

| | | | dst_host_serror_rate > 0.44

| | | | | | flag = SF: normal (3.0)

| | | | | | flag = S0: neptune (2.0/1.0)

| | | | | | flag = REJ: portsweep (2.0)

| | | | | | flag = RSTR: nmap (0.0)

| | | | | | flag = SH: nmap (257.0)

| | | | | | flag = RSTO: normal (1.0)

| | | | | | flag = S1: normal (2.0)

| | | | | | flag = RSTOS0: nmap (0.0)

| | | | | | flag = S3: nmap (0.0)

| | | | | | flag = S2: nmap (0.0)

| | | | | | flag = OTH: nmap (0.0)

| | | protocol_type = udp

| | | | dst_host_diff_srv_rate <= 0.01: rootkit (2.0)

| | | | dst_host_diff_srv_rate > 0.01: satan (61.0)

| | | protocol_type = icmp

| | | | dst_host_srv_diff_host_rate <= 0.12

| | | | | dst_host_rerror_rate <= 0

| | | | | | dst_host_diff_srv_rate <= 0.21: ipsweep (50.0/19.0)

| | | | | | dst_host_diff_srv_rate > 0.21: nmap (6.0)

| | | | | dst_host_rerror_rate > 0: portsweep (2.0)

| | | | dst_host_srv_diff_host_rate > 0.12: nmap (956.0)

| | dst_host_srv_diff_host_rate > 0.48

| | | dst_host_srv_serror_rate <= 0.03

| | | | dst_bytes <= 12

| | | | | dst_host_rerror_rate <= 0.99: ipsweep (2815.0/8.0)

| | | | | dst_host_rerror_rate > 0.99

| | | | | | dst_host_diff_srv_rate <= 0.5: normal (8.0)

| | | | | | dst_host_diff_srv_rate > 0.5: ipsweep (12.0)

| | | | dst_bytes > 12

| | | | | dst_host_rerror_rate <= 0.35: normal (25.0/1.0)

| | | | | dst_host_rerror_rate > 0.35: ipsweep (20.0)

| | | dst_host_srv_serror_rate > 0.03

| | | | land <= 0: normal (14.0/1.0)

| | | | land > 0: land (13.0/5.0)

| count > 2

| | src_bytes <= 0

| | | dst_host_diff_srv_rate <= 0

| | | | dst_host_srv_serror_rate <= 0.33

| | | | | service = ftp_data

| | | | | | same_srv_rate <= 0.87: loadmodule (2.0)

| | | | | | same_srv_rate > 0.87: buffer_overflow (5.0/1.0)

| | | | | service = other: normal (0.0)

| | | | | service = private: neptune (9.0)

| | | | | service = http: normal (286.0)

| | | | | service = remote_job: normal (0.0)

| | | | | service = name: normal (0.0)

| | | | | service = netbios_ns: normal (0.0)

| | | | | service = eco_i: normal (0.0)

| | | | | service = mtp: normal (0.0)

| | | | | service = telnet: normal (0.0)

| | | | | service = finger: land (1.0)

| | | | | service = domain_u: normal (0.0)

| | | | | service = supdup: normal (0.0)

| | | | | service = uucp_path: normal (0.0)

| | | | | service = Z39_50: normal (0.0)

| | | | | service = smtp: normal (0.0)

| | | | | service = csnet_ns: normal (0.0)

| | | | | service = uucp: normal (0.0)

| | | | | service = netbios_dgm: normal (0.0)

| | | | | service = urp_i: normal (0.0)

| | | | | service = auth: normal (0.0)

| | | | | service = domain: normal (0.0)

| | | | | service = ftp: normal (0.0)

| | | | | service = bgp: normal (0.0)

| | | | | service = ldap: normal (0.0)

| | | | | service = ecr_i: normal (0.0)

| | | | | service = gopher: normal (0.0)

| | | | | service = vmnet: normal (0.0)

| | | | | service = systat: normal (0.0)

| | | | | service = http_443: normal (0.0)

| | | | | service = efs: normal (0.0)

| | | | | service = whois: normal (0.0)

| | | | | service = imap4: normal (0.0)

| | | | | service = iso_tsap: normal (0.0)

| | | | | service = echo: normal (0.0)

| | | | | service = klogin: normal (0.0)

| | | | | service = link: normal (0.0)

| | | | | service = sunrpc: normal (0.0)

| | | | | service = login: normal (0.0)

| | | | | service = kshell: normal (0.0)

| | | | | service = sql_net: normal (0.0)

| | | | | service = time: normal (0.0)

| | | | | service = hostnames: normal (0.0)

| | | | | service = exec: normal (0.0)

| | | | | service = ntp_u: normal (0.0)

| | | | | service = discard: normal (0.0)

| | | | | service = nntp: normal (0.0)

| | | | | service = courier: normal (0.0)

| | | | | service = ctf: normal (0.0)

| | | | | service = ssh: normal (0.0)

| | | | | service = daytime: normal (0.0)

| | | | | service = shell: normal (0.0)

| | | | | service = netstat: normal (0.0)

| | | | | service = pop_3: normal (0.0)

| | | | | service = nnsp: normal (0.0)

| | | | | service = IRC: normal (0.0)

| | | | | service = pop_2: normal (0.0)

| | | | | service = printer: normal (0.0)

| | | | | service = tim_i: normal (0.0)

| | | | | service = pm_dump: normal (0.0)

| | | | | service = red_i: normal (0.0)

| | | | | service = netbios_ssn: normal (0.0)

| | | | | service = rje: normal (0.0)

| | | | | service = X11: normal (0.0)

| | | | | service = urh_i: normal (0.0)

| | | | | service = http_8001: normal (0.0)

| | | | | service = aol: normal (0.0)

| | | | | service = http_2784: normal (0.0)

| | | | | service = tftp_u: normal (0.0)

| | | | | service = harvest: normal (0.0)

| | | | dst_host_srv_serror_rate > 0.33

| | | | | flag = SF: imap (4.0)

| | | | | flag = S0: neptune (52.0)

| | | | | flag = REJ: neptune (0.0)

| | | | | flag = RSTR: neptune (0.0)

| | | | | flag = SH: neptune (0.0)

| | | | | flag = RSTO: neptune (0.0)

| | | | | flag = S1: neptune (0.0)

| | | | | flag = RSTOS0: neptune (0.0)

| | | | | flag = S3: neptune (0.0)

| | | | | flag = S2: neptune (0.0)

| | | | | flag = OTH: neptune (0.0)

| | | dst_host_diff_srv_rate > 0

| | | | dst_host_same_src_port_rate <= 0.03

| | | | | diff_srv_rate <= 0.48

| | | | | | diff_srv_rate <= 0.02

| | | | | | | serror_rate <= 0.5

| | | | | | | | dst_host_diff_srv_rate <= 0.04: normal (16.0)

| | | | | | | | dst_host_diff_srv_rate > 0.04: neptune (7.0)

| | | | | | | serror_rate > 0.5: neptune (267.0)

| | | | | | diff_srv_rate > 0.02

| | | | | | | count <= 5

| | | | | | | | serror_rate <= 0.5: normal (3.0)

| | | | | | | | serror_rate > 0.5: neptune (29.0)

| | | | | | | count > 5: neptune (40282.0/1.0)

| | | | | diff_srv_rate > 0.48

| | | | | | dst_host_serror_rate <= 0.5

| | | | | | | dst_host_diff_srv_rate <= 0.04

| | | | | | | | service = ftp_data: normal (1.0)

| | | | | | | | service = other: normal (0.0)

| | | | | | | | service = private: portsweep (7.0)

| | | | | | | | service = http: normal (0.0)

| | | | | | | | service = remote_job: normal (0.0)

| | | | | | | | service = name: normal (0.0)

| | | | | | | | service = netbios_ns: normal (0.0)

| | | | | | | | service = eco_i: normal (0.0)

| | | | | | | | **service = mtp: normal (0.0)**
| | | | | | | | **service = telnet: normal (2.0)**
| | | | | | | | **service = finger: normal (0.0)**
| | | | | | | | **service = domain_u: normal (0.0)**
| | | | | | | | **service = supdup: normal (0.0)**
| | | | | | | | **service = uucp_path: normal (0.0)**
| | | | | | | | **service = Z39_50: normal (0.0)**
| | | | | | | | **service = smtp: normal (1.0)**
| | | | | | | | **service = csnet_ns: normal (0.0)**
| | | | | | | | **service = uucp: satan (1.0)**
| | | | | | | | **service = netbios_dgm: normal (0.0)**
| | | | | | | | **service = urp_i: normal (0.0)**
| | | | | | | | **service = auth: normal (0.0)**
| | | | | | | | **service = domain: normal (0.0)**
| | | | | | | | **service = ftp: normal (0.0)**
| | | | | | | | **service = bgp: normal (0.0)**
| | | | | | | | **service = ldap: normal (0.0)**
| | | | | | | | **service = ecr_i: normal (0.0)**
| | | | | | | | **service = gopher: normal (0.0)**
| | | | | | | | **service = vmnet: normal (0.0)**
| | | | | | | | **service = systat: normal (0.0)**
| | | | | | | | **service = http_443: normal (0.0)**
| | | | | | | | **service = efs: normal (0.0)**
| | | | | | | | **service = whois: normal (0.0)**
| | | | | | | | **service = imap4: normal (0.0)**
| | | | | | | | **service = iso_tsap: normal (0.0)**
| | | | | | | | **service = echo: normal (0.0)**
| | | | | | | | **service = klogin: normal (0.0)**
| | | | | | | | **service = link: normal (0.0)**
| | | | | | | | **service = sunrpc: normal (0.0)**
| | | | | | | | **service = login: normal (0.0)**
| | | | | | | | **service = kshell: normal (0.0)**
| | | | | | | | **service = sql_net: normal (0.0)**
| | | | | | | | **service = time: normal (0.0)**

| | | | | | | | service = hostnames: normal (0.0)

| | | | | | | | service = exec: normal (0.0)

| | | | | | | | service = ntp_u: normal (0.0)

| | | | | | | | service = discard: normal (0.0)

| | | | | | | | service = nntp: normal (0.0)

| | | | | | | | service = courier: normal (0.0)

| | | | | | | | service = ctf: normal (0.0)

| | | | | | | | service = ssh: normal (0.0)

| | | | | | | | service = daytime: normal (0.0)

| | | | | | | | service = shell: normal (0.0)

| | | | | | | | service = netstat: portsweep (2.0)

| | | | | | | | service = pop_3: normal (0.0)

| | | | | | | | service = nnsp: normal (0.0)

| | | | | | | | service = IRC: normal (0.0)

| | | | | | | | service = pop_2: normal (0.0)

| | | | | | | | service = printer: normal (0.0)

| | | | | | | | service = tim_i: normal (0.0)

| | | | | | | | service = pm_dump: normal (0.0)

| | | | | | | | service = red_i: normal (0.0)

| | | | | | | | service = netbios_ssn: normal (0.0)

| | | | | | | | service = rje: normal (0.0)

| | | | | | | | service = X11: normal (6.0/1.0)

| | | | | | | | service = urh_i: normal (0.0)

| | | | | | | | service = http_8001: normal (0.0)

| | | | | | | | service = aol: normal (0.0)

| | | | | | | | service = http_2784: normal (0.0)

| | | | | | | | service = tftp_u: normal (0.0)

| | | | | | | | service = harvest: normal (0.0)

| | | | | | | dst_host_diff_srv_rate > 0.04: satan (2052.0)

| | | | | | dst_host_serror_rate > 0.5: neptune (214.0/3.0)

| | | | dst_host_same_src_port_rate > 0.03

| | | | | diff_srv_rate <= 0.33: neptune (216.0/4.0)

| | | | | diff_srv_rate > 0.33: portsweep (461.0/9.0)

| | src_bytes > 0

| | | src_bytes <= 6: satan (1439.0/11.0)

| | | src_bytes > 6

| | | | protocol_type = tcp: normal (6.0/1.0)

| | | | protocol_type = udp: normal (0.0)

| | | | protocol_type = icmp: ipsweep (5.0/1.0)

src_bytes > 8

| wrong_fragment <= 0

| | src_bytes <= 16787

| | | dst_host_srv_diff_host_rate <= 0.1

| | | | dst_bytes <= 0

| | | | | service = ftp_data

| | | | | | src_bytes <= 353

| | | | | | | src_bytes <= 326: normal (1416.0/10.0)

| | | | | | | src_bytes > 326: warezclient (101.0/1.0)

| | | | | | src_bytes > 353: normal (2898.0/20.0)

| | | | | service = other: normal (687.0/3.0)

| | | | | service = private

| | | | | | src_bytes <= 156

| | | | | | | src_bytes <= 102: nmap (40.0/1.0)

| | | | | | | src_bytes > 102: normal (119.0)

| | | | | | src_bytes > 156: nmap (211.0/5.0)

| | | | | service = http: normal (59.0)

| | | | | service = remote_job: normal (0.0)

| | | | | service = name: normal (0.0)

| | | | | service = netbios_ns: normal (0.0)

| | | | | service = eco_i

| | | | | | src_bytes <= 25

| | | | | | | src_bytes <= 19: ipsweep (6.0)

| | | | | | | src_bytes > 19: satan (20.0)

| | | | | | src_bytes > 25: normal (347.0)

| | | | | service = mtp: normal (0.0)

| | | | | service = telnet: normal (0.0)

| | | | | service = finger: normal (0.0)

| | | | | service = domain_u: normal (479.0)

| | | | | service = supdup: normal (0.0)
| | | | | service = uucp_path: normal (0.0)
| | | | | service = Z39_50: normal (0.0)
| | | | | service = smtp: normal (6.0)
| | | | | service = csnet_ns: normal (0.0)
| | | | | service = uucp: normal (0.0)
| | | | | service = netbios_dgm: normal (0.0)
| | | | | service = urp_i
| | | | | | src_bytes <= 50: satan (3.0)
| | | | | | src_bytes > 50: normal (589.0)
| | | | | service = auth: normal (0.0)
| | | | | service = domain: normal (0.0)
| | | | | service = ftp: normal (0.0)
| | | | | service = bgp: normal (0.0)
| | | | | service = ldap: normal (0.0)
| | | | | service = ecr_i
| | | | | | src_bytes <= 292
| | | | | | | src_bytes <= 25
| | | | | | | | src_bytes <= 19: ipsweep (16.0)
| | | | | | | | src_bytes > 19: satan (10.0/1.0)
| | | | | | | src_bytes > 25: normal (178.0)
| | | | | | src_bytes > 292: smurf (2646.0)
| | | | | service = gopher: normal (0.0)
| | | | | service = vmnet: normal (0.0)
| | | | | service = systat: normal (0.0)
| | | | | service = http_443: normal (0.0)
| | | | | service = efs: normal (0.0)
| | | | | service = whois: normal (0.0)
| | | | | service = imap4: normal (0.0)
| | | | | service = iso_tsap: normal (0.0)
| | | | | service = echo: normal (0.0)
| | | | | service = klogin: normal (0.0)
| | | | | service = link: normal (0.0)
| | | | | service = sunrpc: normal (0.0)

| | | | | service = login: normal (0.0)

| | | | | service = kshell: normal (0.0)

| | | | | service = sql_net: normal (0.0)

| | | | | service = time: normal (0.0)

| | | | | service = hostnames: normal (0.0)

| | | | | service = exec: normal (0.0)

| | | | | service = ntp_u: normal (0.0)

| | | | | service = discard: normal (0.0)

| | | | | service = nntp: normal (0.0)

| | | | | service = courier: normal (0.0)

| | | | | service = ctf: normal (0.0)

| | | | | service = ssh: normal (0.0)

| | | | | service = daytime: normal (0.0)

| | | | | service = shell: normal (0.0)

| | | | | service = netstat: normal (0.0)

| | | | | service = pop_3: normal (0.0)

| | | | | service = nnsp: normal (0.0)

| | | | | service = IRC: normal (0.0)

| | | | | service = pop_2: normal (0.0)

| | | | | service = printer: normal (0.0)

| | | | | service = tim_i

| | | | | | dst_host_diff_srv_rate <= 0.01: pod (4.0/1.0)

| | | | | | dst_host_diff_srv_rate > 0.01: normal (4.0)

| | | | | service = pm_dump: normal (0.0)

| | | | | service = red_i: normal (8.0)

| | | | | service = netbios_ssn: normal (0.0)

| | | | | service = rje: normal (0.0)

| | | | | service = X11: normal (0.0)

| | | | | service = urh_i: normal (10.0)

| | | | | service = http_8001: normal (0.0)

| | | | | service = aol: normal (0.0)

| | | | | service = http_2784: normal (0.0)

| | | | | service = tftp_u: normal (0.0)

| | | | | service = harvest: normal (0.0)

| | | | dst_bytes > 0

| | | | | hot <= 0

| | | | | | diff_srv_rate <= 0.69: normal (53255.0/41.0)

| | | | | | diff_srv_rate > 0.69

| | | | | | | count <= 4: normal (449.0/7.0)

| | | | | | | count > 4: satan (9.0)

| | | | | hot > 0

| | | | | | hot <= 25

| | | | | | | src_bytes <= 1551

| | | | | | | | src_bytes <= 130

| | | | | | | | | dst_host_diff_srv_rate <= 0.01

| | | | | | | | | | service = ftp_data: guess_passwd (0.0)

| | | | | | | | | | service = other: guess_passwd (0.0)

| | | | | | | | | | service = private: guess_passwd (0.0)

| | | | | | | | | | service = http: phf (4.0)

| | | | | | | | | | service = remote_job: guess_passwd (0.0)

| | | | | | | | | | service = name: guess_passwd (0.0)

| | | | | | | | | | service = netbios_ns: guess_passwd (0.0)

| | | | | | | | | | service = eco_i: guess_passwd (0.0)

| | | | | | | | | | service = mtp: guess_passwd (0.0)

| | | | | | | | | | service = telnet: guess_passwd (52.0)

| | | | | | | | | | service = finger: guess_passwd (0.0)

| | | | | | | | | | service = domain_u: guess_passwd (0.0)

| | | | | | | | | | service = supdup: guess_passwd (0.0)

| | | | | | | | | | service = uucp_path: guess_passwd (0.0)

| | | | | | | | | | service = Z39_50: guess_passwd (0.0)

| | | | | | | | | | service = smtp: guess_passwd (0.0)

| | | | | | | | | | service = csnet_ns: guess_passwd (0.0)

| | | | | | | | | | service = uucp: guess_passwd (0.0)

| | | | | | | | | | service = netbios_dgm: guess_passwd (0.0)

| | | | | | | | | | service = urp_i: guess_passwd (0.0)

| | | | | | | | | | service = auth: guess_passwd (0.0)

| | | | | | | | | | service = domain: guess_passwd (0.0)

| | | | | | | | | | service = ftp

| | | | | | | | | | | dst_bytes <= 437: multihop (2.0)

| | | | | | | | | | | dst_bytes > 437: ftp_write (2.0)

| | | | | | | | | service = bgp: guess_passwd (0.0)

| | | | | | | | | service = ldap: guess_passwd (0.0)

| | | | | | | | | service = ecr_i: guess_passwd (0.0)

| | | | | | | | | service = gopher: guess_passwd (0.0)

| | | | | | | | | service = vmnet: guess_passwd (0.0)

| | | | | | | | | service = systat: guess_passwd (0.0)

| | | | | | | | | service = http_443: guess_passwd (0.0)

| | | | | | | | | service = efs: guess_passwd (0.0)

| | | | | | | | | service = whois: guess_passwd (0.0)

| | | | | | | | | service = imap4: guess_passwd (0.0)

| | | | | | | | | service = iso_tsap: guess_passwd (0.0)

| | | | | | | | | service = echo: guess_passwd (0.0)

| | | | | | | | | service = klogin: guess_passwd (0.0)

| | | | | | | | | service = link: guess_passwd (0.0)

| | | | | | | | | service = sunrpc: guess_passwd (0.0)

| | | | | | | | | service = login: guess_passwd (0.0)

| | | | | | | | | service = kshell: guess_passwd (0.0)

| | | | | | | | | service = sql_net: guess_passwd (0.0)

| | | | | | | | | service = time: guess_passwd (0.0)

| | | | | | | | | service = hostnames: guess_passwd (0.0)

| | | | | | | | | service = exec: guess_passwd (0.0)

| | | | | | | | | service = ntp_u: guess_passwd (0.0)

| | | | | | | | | service = discard: guess_passwd (0.0)

| | | | | | | | | service = nntp: guess_passwd (0.0)

| | | | | | | | | service = courier: guess_passwd (0.0)

| | | | | | | | | service = ctf: guess_passwd (0.0)

| | | | | | | | | service = ssh: guess_passwd (0.0)

| | | | | | | | | service = daytime: guess_passwd (0.0)

| | | | | | | | | service = shell: guess_passwd (0.0)

| | | | | | | | | service = netstat: guess_passwd (0.0)

| | | | | | | | | service = pop_3: guess_passwd (0.0)

| | | | | | | | | service = nnsp: guess_passwd (0.0)

| | | | | | | | | service = IRC: guess_passwd (0.0)
| | | | | | | | | service = pop_2: guess_passwd (0.0)
| | | | | | | | | service = printer: guess_passwd (0.0)
| | | | | | | | | service = tim_i: guess_passwd (0.0)
| | | | | | | | | service = pm_dump: guess_passwd (0.0)
| | | | | | | | | service = red_i: guess_passwd (0.0)
| | | | | | | | | service = netbios_ssn: guess_passwd (0.0)
| | | | | | | | | service = rje: guess_passwd (0.0)
| | | | | | | | | service = X11: guess_passwd (0.0)
| | | | | | | | | service = urh_i: guess_passwd (0.0)
| | | | | | | | | service = http_8001: guess_passwd (0.0)
| | | | | | | | | service = aol: guess_passwd (0.0)
| | | | | | | | | service = http_2784: guess_passwd (0.0)
| | | | | | | | | service = tftp_u: guess_passwd (0.0)
| | | | | | | | | service = harvest: guess_passwd (0.0)
| | | | | | | | dst_host_diff_srv_rate > 0.01
| | | | | | | | | logged_in <= 0
| | | | | | | | | | hot <= 1: satan (4.0)
| | | | | | | | | | hot > 1: normal (2.0)
| | | | | | | | | logged_in > 0: normal (8.0/1.0)
| | | | | | | src_bytes > 130
| | | | | | | | dst_host_serror_rate <= 0.1: normal (867.0/14.0)
| | | | | | | | dst_host_serror_rate > 0.1
| | | | | | | | | hot <= 2: normal (18.0)
| | | | | | | | | hot > 2
| | | | | | | | | | service = ftp_data: warezclient (0.0)
| | | | | | | | | | service = other: warezclient (0.0)
| | | | | | | | | | service = private: warezclient (0.0)
| | | | | | | | | | service = http: warezclient (0.0)
| | | | | | | | | | service = remote_job: warezclient (0.0)
| | | | | | | | | | service = name: warezclient (0.0)
| | | | | | | | | | service = netbios_ns: warezclient (0.0)
| | | | | | | | | | service = eco_i: warezclient (0.0)
| | | | | | | | | | service = mtp: warezclient (0.0)

| | | | | | | | | | | **service = telnet: normal (2.0)**
| | | | | | | | | | | **service = finger: warezclient (0.0)**
| | | | | | | | | | | **service = domain_u: warezclient (0.0)**
| | | | | | | | | | | **service = supdup: warezclient (0.0)**
| | | | | | | | | | | **service = uucp_path: warezclient (0.0)**
| | | | | | | | | | | **service = Z39_50: warezclient (0.0)**
| | | | | | | | | | | **service = smtp: warezclient (0.0)**
| | | | | | | | | | | **service = csnet_ns: warezclient (0.0)**
| | | | | | | | | | | **service = uucp: warezclient (0.0)**
| | | | | | | | | | | **service = netbios_dgm: warezclient (0.0)**
| | | | | | | | | | | **service = urp_i: warezclient (0.0)**
| | | | | | | | | | | **service = auth: warezclient (0.0)**
| | | | | | | | | | | **service = domain: warezclient (0.0)**
| | | | | | | | | | | **service = ftp: warezclient (27.0)**
| | | | | | | | | | | **service = bgp: warezclient (0.0)**
| | | | | | | | | | | **service = ldap: warezclient (0.0)**
| | | | | | | | | | | **service = ecr_i: warezclient (0.0)**
| | | | | | | | | | | **service = gopher: warezclient (0.0)**
| | | | | | | | | | | **service = vmnet: warezclient (0.0)**
| | | | | | | | | | | **service = systat: warezclient (0.0)**
| | | | | | | | | | | **service = http_443: warezclient (0.0)**
| | | | | | | | | | | **service = efs: warezclient (0.0)**
| | | | | | | | | | | **service = whois: warezclient (0.0)**
| | | | | | | | | | | **service = imap4: warezclient (0.0)**
| | | | | | | | | | | **service = iso_tsap: warezclient (0.0)**
| | | | | | | | | | | **service = echo: warezclient (0.0)**
| | | | | | | | | | | **service = klogin: warezclient (0.0)**
| | | | | | | | | | | **service = link: warezclient (0.0)**
| | | | | | | | | | | **service = sunrpc: warezclient (0.0)**
| | | | | | | | | | | **service = login: warezclient (0.0)**
| | | | | | | | | | | **service = kshell: warezclient (0.0)**
| | | | | | | | | | | **service = sql_net: warezclient (0.0)**
| | | | | | | | | | | **service = time: warezclient (0.0)**
| | | | | | | | | | | **service = hostnames: warezclient (0.0)**

| | | | | | | | | | service = exec: warezclient (0.0)

| | | | | | | | | | service = ntp_u: warezclient (0.0)

| | | | | | | | | | service = discard: warezclient (0.0)

| | | | | | | | | | service = nntp: warezclient (0.0)

| | | | | | | | | | service = courier: warezclient (0.0)

| | | | | | | | | | service = ctf: warezclient (0.0)

| | | | | | | | | | service = ssh: warezclient (0.0)

| | | | | | | | | | service = daytime: warezclient (0.0)

| | | | | | | | | | service = shell: warezclient (0.0)

| | | | | | | | | | service = netstat: warezclient (0.0)

| | | | | | | | | | service = pop_3: warezclient (0.0)

| | | | | | | | | | service = nnsp: warezclient (0.0)

| | | | | | | | | | service = IRC: warezclient (0.0)

| | | | | | | | | | service = pop_2: warezclient (0.0)

| | | | | | | | | | service = printer: warezclient (0.0)

| | | | | | | | | | service = tim_i: warezclient (0.0)

| | | | | | | | | | service = pm_dump: warezclient (0.0)

| | | | | | | | | | service = red_i: warezclient (0.0)

| | | | | | | | | | service = netbios_ssn: warezclient (0.0)

| | | | | | | | | | service = rje: warezclient (0.0)

| | | | | | | | | | service = X11: warezclient (0.0)

| | | | | | | | | | service = urh_i: warezclient (0.0)

| | | | | | | | | | service = http_8001: warezclient (0.0)

| | | | | | | | | | service = aol: warezclient (0.0)

| | | | | | | | | | service = http_2784: warezclient (0.0)

| | | | | | | | | | service = tftp_u: warezclient (0.0)

| | | | | | | | | | service = harvest: warezclient (0.0)

| | | | | | | src_bytes > 1551

| | | | | | | | dst_host_diff_srv_rate <= 0

| | | | | | | | | service = ftp_data: buffer_overflow (0.0)

| | | | | | | | | service = other: buffer_overflow (0.0)

| | | | | | | | | service = private: buffer_overflow (0.0)

| | | | | | | | | service = http: back (4.0)

| | | | | | | | | service = remote_job: buffer_overflow (0.0)

| | | | | | | | | **service = name: buffer_overflow (0.0)**
| | | | | | | | | **service = netbios_ns: buffer_overflow (0.0)**
| | | | | | | | | **service = eco_i: buffer_overflow (0.0)**
| | | | | | | | | **service = mtp: buffer_overflow (0.0)**
| | | | | | | | | **service = telnet: buffer_overflow (15.0)**
| | | | | | | | | **service = finger: buffer_overflow (0.0)**
| | | | | | | | | **service = domain_u: buffer_overflow (0.0)**
| | | | | | | | | **service = supdup: buffer_overflow (0.0)**
| | | | | | | | | **service = uucp_path: buffer_overflow (0.0)**
| | | | | | | | | **service = Z39_50: buffer_overflow (0.0)**
| | | | | | | | | **service = smtp: buffer_overflow (0.0)**
| | | | | | | | | **service = csnet_ns: buffer_overflow (0.0)**
| | | | | | | | | **service = uucp: buffer_overflow (0.0)**
| | | | | | | | | **service = netbios_dgm: buffer_overflow (0.0)**
| | | | | | | | | **service = urp_i: buffer_overflow (0.0)**
| | | | | | | | | **service = auth: buffer_overflow (0.0)**
| | | | | | | | | **service = domain: buffer_overflow (0.0)**
| | | | | | | | | **service = ftp: buffer_overflow (0.0)**
| | | | | | | | | **service = bgp: buffer_overflow (0.0)**
| | | | | | | | | **service = ldap: buffer_overflow (0.0)**
| | | | | | | | | **service = ecr_i: buffer_overflow (0.0)**
| | | | | | | | | **service = gopher: buffer_overflow (0.0)**
| | | | | | | | | **service = vmnet: buffer_overflow (0.0)**
| | | | | | | | | **service = systat: buffer_overflow (0.0)**
| | | | | | | | | **service = http_443: buffer_overflow (0.0)**
| | | | | | | | | **service = efs: buffer_overflow (0.0)**
| | | | | | | | | **service = whois: buffer_overflow (0.0)**
| | | | | | | | | **service = imap4: buffer_overflow (0.0)**
| | | | | | | | | **service = iso_tsap: buffer_overflow (0.0)**
| | | | | | | | | **service = echo: buffer_overflow (0.0)**
| | | | | | | | | **service = klogin: buffer_overflow (0.0)**
| | | | | | | | | **service = link: buffer_overflow (0.0)**
| | | | | | | | | **service = sunrpc: buffer_overflow (0.0)**
| | | | | | | | | **service = login: buffer_overflow (0.0)**

| | | | | | | | | service = kshell: buffer_overflow (0.0)
| | | | | | | | | service = sql_net: buffer_overflow (0.0)
| | | | | | | | | service = time: buffer_overflow (0.0)
| | | | | | | | | service = hostnames: buffer_overflow (0.0)
| | | | | | | | | service = exec: buffer_overflow (0.0)
| | | | | | | | | service = ntp_u: buffer_overflow (0.0)
| | | | | | | | | service = discard: buffer_overflow (0.0)
| | | | | | | | | service = nntp: buffer_overflow (0.0)
| | | | | | | | | service = courier: buffer_overflow (0.0)
| | | | | | | | | service = ctf: buffer_overflow (0.0)
| | | | | | | | | service = ssh: buffer_overflow (0.0)
| | | | | | | | | service = daytime: buffer_overflow (0.0)
| | | | | | | | | service = shell: buffer_overflow (0.0)
| | | | | | | | | service = netstat: buffer_overflow (0.0)
| | | | | | | | | service = pop_3: buffer_overflow (0.0)
| | | | | | | | | service = nnsp: buffer_overflow (0.0)
| | | | | | | | | service = IRC: buffer_overflow (0.0)
| | | | | | | | | service = pop_2: buffer_overflow (0.0)
| | | | | | | | | service = printer: buffer_overflow (0.0)
| | | | | | | | | service = tim_i: buffer_overflow (0.0)
| | | | | | | | | service = pm_dump: buffer_overflow (0.0)
| | | | | | | | | service = red_i: buffer_overflow (0.0)
| | | | | | | | | service = netbios_ssn: buffer_overflow (0.0)
| | | | | | | | | service = rje: buffer_overflow (0.0)
| | | | | | | | | service = X11: buffer_overflow (0.0)
| | | | | | | | | service = urh_i: buffer_overflow (0.0)
| | | | | | | | | service = http_8001: buffer_overflow (0.0)
| | | | | | | | | service = aol: buffer_overflow (0.0)
| | | | | | | | | service = http_2784: buffer_overflow (0.0)
| | | | | | | | | service = tftp_u: buffer_overflow (0.0)
| | | | | | | | | service = harvest: buffer_overflow (0.0)
| | | | | | | | dst_host_diff_srv_rate > 0: normal (16.0)
| | | | | | hot > 25
| | | | | | | dst_bytes <= 3299: warezclient (273.0)

| | | | | | | dst_bytes > 3299: normal (262.0)

| | | dst_host_srv_diff_host_rate > 0.1

| | | | protocol_type = tcp

| | | | | dst_host_same_src_port_rate <= 0.59: normal (1021.0/8.0)

| | | | | dst_host_same_src_port_rate > 0.59

| | | | | | dst_bytes <= 6

| | | | | | | logged_in <= 0: normal (4.0/1.0)

| | | | | | | logged_in > 0

| | | | | | | | dst_host_same_src_port_rate <= 0.82

| | | | | | | | | src_bytes <= 326: normal (5.0)

| | | | | | | | | src_bytes > 326

| | | | | | | | | | src_bytes <= 593: warezclient (7.0)

| | | | | | | | | | src_bytes > 593: normal (2.0)

| | | | | | | | dst_host_same_src_port_rate > 0.82: warezclient (386.0/1.0)

| | | | | | dst_bytes > 6

| | | | | | | hot <= 1: normal (110.0/1.0)

| | | | | | | hot > 1

| | | | | | | | hot <= 2: normal (2.0/1.0)

| | | | | | | | hot > 2: buffer_overflow (2.0)

| | | | protocol_type = udp: normal (34.0)

| | | | protocol_type = icmp

| | | | | src_bytes <= 24: ipsweep (530.0)

| | | | | src_bytes > 24: normal (171.0)

| | src_bytes > 16787

| | | hot <= 0

| | | | dst_host_diff_srv_rate <= 0

| | | | | dst_host_same_src_port_rate <= 0.03: back (7.0)

| | | | | dst_host_same_src_port_rate > 0.03

| | | | | | src_bytes <= 2293136: normal (16.0/1.0)

| | | | | | src_bytes > 2293136: warezclient (9.0/1.0)

| | | | dst_host_diff_srv_rate > 0

| | | | | src_bytes <= 14584085: normal (640.0)

| | | | | src_bytes > 14584085

| | | | | | logged_in <= 0: portsweep (7.0)

| | | | | | logged_in > 0: normal (5.0)

| | | hot > 0

| | | | dst_bytes <= 730: warezclient (53.0/1.0)

| | | | dst_bytes > 730: back (945.0/2.0)

| wrong_fragment > 0

| | protocol_type = tcp: teardrop (0.0)

| | protocol_type = udp: teardrop (892.0)

| | protocol_type = icmp: pod (198.0)

Number of Leaves : 698

Size of the tree : 811

Summary of appendix A:

SIDDM model developed in this research, uses above tree for classification purpose. Decision tree generated in this research uses j48 decision tree with appropriate parameter setting. This tree shows features and associated values of features which decide whether it is attack data or normal. This annexure shows result of experiments.

Publications By Researcher based on this Thesis

International Journal Paper

- "Investigation of feature selection and ensemble methods for performance improvement of intrusion attack classification" In International journal of computer science and Engineering (IJCSE), U.S.A, ISSN 2278-9960 . (JUL 2013)
- "Efficient algorithm for intrusion attack classification by analyzing KDD cup 99" , ISSN 978-1-4673-1989-8/12 **IEEE Xplore™** .(SEP 2012)
- "Comparative analysis of two algorithm for intrusion attack classification using KDD cup dataset" In International journal of computer science and Engineering , Roseville, U.S.A ,ISSN 2278-9979.(AUG 2012)
- "Selection of Relevant Feature for Intrusion Attack Classification by Analyzing KDD Cup99" ,in Moradabad Institute of Technology International Journal of Computer Science & Information Technology (MITIJCSIT) ISSN 2230 7621, eISSN 2230 763X(AUG 2012) .

International Conference Paper

- "Data mining solution for analysis of intrusion based security attack" in The 7th International Conference on IT Applications and Management: Technological Innovation and the Future of Culture and Tourism organized by Institute of Management, JK Lakshmipat University , Jaipur In association with The Korea Database Society, Hanyang University ,Seoul ,Korea (Dec 2011) .

Annexure 3 Publications by Researcher based on this Thesis

Citation of papers

Related to this research work on

Google Scholar