for Intrusion Detection System (IDS) to tackle zero day attacks or novel attacks. While many researches [5][6] focused on the accuracy and timely detection of attacks in the network, none of the studies aims at the robustness of their proposed approach. Here, robustness of an approach indicates how effective it is to detect anomaly in different datasets. To the best knowledge, we, for the first time, investigate in this direction and proposed an approach that can achieve high performance to detect anomaly in different datasets.

The question is, where does the Intrusion detection system fit in the design. To put it in simpler terms, an Intrusion detection system can be compared with a burglar alarm. For example, the lock system in a car protects the car from theft. But if somebody breaks the lock system and tries to steal the car, it is the burglar alarm that detects that the lock has been broken and alerts the owner by raising an alarm. The Intrusion detection system in a similar way complements the firewall security. The firewall protects an organization from malicious attacks from the Internet and the Intrusion detection system detects if someone tries to break in through the firewall or manages to break in the firewall security and tries to have access on any system in the trusted side and alerts the system administrator in case there is a breach in security. Moreover, Firewalls do a very good job of filtering incoming traffic from the Internet; however, there are ways to circumvent the firewall. For example, external users can connect to the Intranet by dialling in through a modem installed in the private network of the organization. This kind of access would not be seen by the firewall. Therefore, an Intrusion detection system (IDS) is a security system that monitors computer systems and network traffic and analyzes that traffic for possible hostile attacks originating from outside the organization and also for system misuse or attacks originating from inside the organization.

1.3 TYPES OF IDS

1.3.1 Network Intrusion Detection System (NIDS)

A Network Intrusion Detection System (NIDS) is one common type of IDS that analyses network traffic at all layers of the Open Systems Interconnection (OSI) model and makes decisions about the purpose of the traffic, analysing for suspicious activity.

1.3.2 The Host Intrusion Detection System (HIDS)

According to the source of the data to examine, the Host Based Intrusion Detection System can be classified in two categories: The HIDS Based Application. The IDS of this type receive the data in application, for example, the logs files generated by the management software of the database, the server web or the firewalls.

1.4 INTRUSION DETECTION METHODS

1.4.1 Signature based intrusion detection system (S-IDS)

This method is for detection of misuse. Behavioural Pattern (signatures) of known attacks are stored in a database. The detection process matches the events pattern against the stored signatures. If a match is found an intrusion signal is generated. It has a major drawback that this method fails to identify new attacks whose patterns are not previously stored or same as known attacks

1.4.2 Anomaly based intrusion detection system (A-IDS)

A-IDS detect unknown or novelty attacks. New or unknown attacks are known as novelty attacks. A-IDS detect such attacks which were not used for training the machine learning system

2 LITERATURE SURVEY

2.1 Intrusion Detection System using AI and Machine Learning Algorithm

There are many types of dangers on the internet, including malware and DDOS attacks. A network can be protected against such attacks using an intrusion detection system. An IDS system can detect intrusions and intrusion de-generates an alert when it detects an intrusion. This intrusion detection system in a network analyzes all traffic. For large datacenters this is a difficult task. There's an enormous amount of data through the network of a data center. Standard intrusion systems cannot then all traffic completely.
A way to fix this is by IP flows is regeneration of packet data. Using IP flows ensures that an intrusion detection system can check all traffic. Intrusion detection systems also require a lot of maintenance. This depends on course and also involves high cost. Sensitive data is also increasingly being stored digitally. All these new services could contain security flaws which could leak private data, such as passwords or other sensitive data. This means that security flaws become more and more important since they can cause so much damage. It is not just the leaking of sensitive data that is an issue, but also protecting a computer or network against malware is important.

Considering this, it becomes more important to be able to detect and prevent attacks on network systems. Intrusion detection systems are used for this purpose. An intrusion detection system can alert administrators of malicious behavior. In order to have good performance, most intrusion detection systems need a lot of manual maintenance. This thesis tries to find out whether an intrusion detection system can work out-of-the-box with an acceptable performance. This is done by using machine learning algorithms. These are algorithms which can learn and find patterns in input. Machine learning algorithms seem promising for the problem of automatic intrusion detection. This thesis tries to view therefore that an intrusion detection system out-of-the-box may have a good performance. This is done via machine learning algorithms. These are algorithms that can learn from data and patterns. This seems well applicable to the problem of intrusion detection, this will also view this thesis, as well as the algorithms may or may not work.

2.2 Attacks Classification

A useful classification is to first make a distinction between internal and external malicious behavior. This makes it easier for humans to understand. The IDS itself can work

with different kind of classifications. However, the IDS have to communicate with an administrator about the detections. A distinction between internal and external malicious behavior is easier to understand. Every type of malicious behavior is identified by different characteristics. Knowing these characteristics is useful to be able to tweak the IDS to make identification more effective.

2.3 External Abnormal Behavior

External abnormal behavior consists of different kind of attacks on the systems. There are much different type of attacks. There are Physical attacks, Buffer overflows, Distributed Denial of Service, Brute- force attacks, Vulnerability scans and Man in the middle attacks.

2.4 Internal Abnormal Behavior

Internal abnormal behavior can be called malware. There are several types of malware. There are four distinct categories of malware. There are Botnets, Viruses Trojan Horses and Worms. Malware are actual programs that infect a system to execute a specific task. The task of the malware defines which category the malware belongs in.

2.5 A look at Firewall

Firewall is like a fence to everybody's computer. It is the first level of security to the Internet. Computer security professionals, Governments, Internet Service Providers, Computer dealers and Manufacturers recommend that everyone must install firewalls, if the computer is connected to the internet.

The important thing to be considered is that firewalls should be properly installed and configured. It should also be properly maintained and updated periodically. Firewall is the first piece of intrusion prevention. It prevents from all kinds of strange attacks. Firewalls are software applications or hardware devices that you install on your system. They are designed to prevent unauthorized access to or from a private network that is connected to the Internet. When a firewall is installed, all incoming or outgoing messages pass through the firewall. Those that do not meet the specified security criteria are blocked by firewall. Firewall provides protection from vulnerable services. It provides controlled access to sites. Most home firewalls are software applications.

There are various types of firewalls, and they work through different processes. However, the following is true for most of the home or personal firewall software that is used today. Information over the Internet is sent in "packets" of data. These packets travel from a source machine to a destination machine -- which could be very near or very far away. Each packet of data contains the IP address and port number of the originating machine. The firewall inspects every packet of data that arrives at the computer -before the data is allowed entry into the system and before it connects with an "open" port. The beauty of a firewall lies in its ability to be selective about what it accepts and what it blocks. The firewall has the ability to refuse any suspect data. If the incoming data is ignored and not allowed in, that port will effectively

disappear on the Internet and hackers cannot find it or connect through it. In other words, instead of receiving a signal that a port is open, the hackers receive nothing back and have no way of connecting.

2.6 Intrusion Detection System (IDS)

Intrusion Detection System has become standard component in security infrastructure as they allow administrators to detect policy violations. Unfortunately, the data collected for analysis is too large, and the analysing process is also time consuming. Today's system consists of multiple node executing multiple Operating Systems that are linked together to form a single distributed system. Today there are many Intrusion Detection System available. Evaluating these IDSs is a difficult task due to various reasons - it is very hard to get high-quality data for performing the evaluation due to privacy and several competitive issues;

-in real time data, labelling network connections as normal or intrusions need a lot of time for experts;

- constant change of network traffic;

- complexity in measuring detection rate and false alarm rate;

- with the types of attacks.

2.6.1 Why do we need IDS?

Intrusions that concern system administrators are

- Modification of system files by unauthorized persons so as to have illegal access to either system or user data;

- Modification of table or other system information in network;

- Unauthorized use of computing resources;

It is a common misunderstanding that firewalls can recognize and block intruders.

A firewall is simply a fence around a network. A fence has neither the capability of detecting somebody trying to break in (such as digging a hole underneath or jumping over it), nor can differentiate somebody carry through the gate is allowed in. A firewall simply restricts access to the designated points in the network. Intrusion Detection System is configured to respond to predefined suspicious activities. An IDS does not replace firewalls. Firewalls are must in any corporate security foundations. Intrusion Detection Systems identify attacks against networks or a host that firewalls is unable to see. Having IDS to complement a firewall can provide an extra layer of protection to a system such as

• Identify attacks that firewall legitimately allow through (such as http attacks against web

servers);

• Identify attempts such as port scan;

• Notice inside hacking;

• Provides additional checks for holes/ports opened through firewalls intentionally or unintentionally.

Intrusion Detection is a set of techniques and methods that are used to detect suspicious activity both at the network and host level. Using Intrusion Detection, we can collect and use information from known types of attacks and find out if someone is trying to attack our network or particular hosts. The information helps us to harden our network security, as well as for legal purposes. Today there are two basic approaches to Intrusion detection. One is anomaly detection and another is misuse detection.

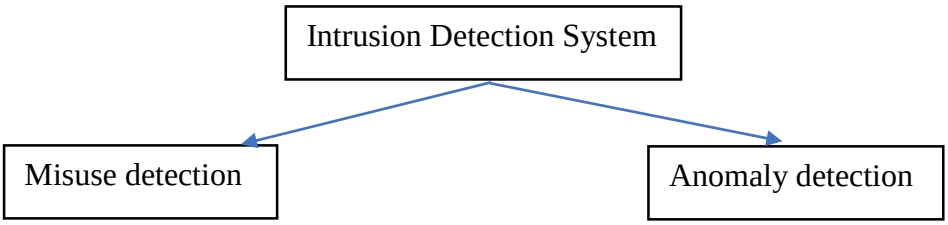


Figure 1. basic types of IDS.

2.6.2 Anomaly detection

This approach identifies deviations from 'normal' behaviour and automatically detects. It observes the behaviour of the system or user for a certain period of time and thereafter declares it as intrusion. It is also called behaviour-based IDS. The advantage of the anomaly IDS is that they can detect new attacks (unknown to the system). They are less dependent on the Operating System specific information. The disadvantage of the anomaly IDS is the generation of a large number of false alarms. Misuse detection also called attacks that are precisely encoded in a manner that captures rearrangements and variations of activities that exploit the same vulnerability. It is based on knowledge about the attacks that were collected. The attacks may be the previous successful one performed to other systems. This attack information may be written as set of rules/policies in defining the IDS. It is also called as signature-based IDS or knowledge-based IDS.

The advantages of misuse IDS is that, they have low false alarm rate. The analysis process of alarms is easier for Network Security administrator as the rules/policies are easy to understand and react quickly.

The disadvantage of misuse IDS is that keeping the knowledge base of such intrusion detection system up to date is not easy. Even after gathering information about the attacks it is time consuming to analyse them and update the knowledge base of IDS. Another disadvantage is generalization of the Intrusion Detection System (IDS) because most of the attacks are dependent on the Operating system, version, platform, and application. Sometimes, a distinction is made between misuse and intrusion detection. The term intrusion is used to describe attack from outside environment; whereas, misuse is used to describe an attack that originates from the local network (internal attack). Intrusion Detection Systems that operate on a host to detect malicious activity are called Host based Intrusion Detection Systems (HIDS), and Intrusion Detection Systems that operate on network are called Network Intrusion Detection Systems (NIDS). As network attacks grows in severity and sophistication, Collaborative Intrusion Detection systems (CIDS)[36] have been attracted much interest today. Collecting data from multiple points in the Internet is essential for correlating malicious activity and extracting robust attack signatures.

Important features an IDS should possess are

• It should be fault tolerant and run continually with minimal human supervision. The IDS must be able to recover from system crashes.

• It should possess the ability to resist subversion so that an attacker cannot disable or modify the IDS easily

• It should have minimal overhead on the system to avoid interfering with the normal operation of the system.

• It should be adaptable to changes in system and user behaviour over time. • It should be portable to different architecture and Operating Systems through simple installation and mechanisms and also easy to use by operators.

• It should be able to detect different types of attacks and must not identify any legitimate activity as an attack (false positives).

• It should not fail to identify any real attacks (false negatives).

2.6.3 Efficiency of The Intrusion Detection System consists of the following:

1) Accuracy: Accuracy deals with the proper detection of the intrusions and absence of false alarms. Inaccuracy occurs when the intrusion detection system reports non-intrusive actions as intrusive.

2) Performance: The performance of the intrusion detection system depends on the rate at which it processes the information. If this rate is too low then the real time sniffing is likely to be not possible.

3) Completeness: The capability of the intrusion detection system to detect all the attacks is called completeness of the system.

4) Fault tolerance: Intrusion detection systems should be resistant to any kind of attacks from the intruders. i.e., IDS should not succumb to an attack.

5) Timeliness: Intrusion detection systems should react analyse and report the systems security officers as quick as possible, in order to let them give time to react before the attack is completely performed.

Intrusion detection systems according to their data source Intrusion detection System can be implemented by study of data sources got from various log files.

Intrusion detection systems according to their data source.

Intrusion detection System can be implemented by study of data sources got from various log

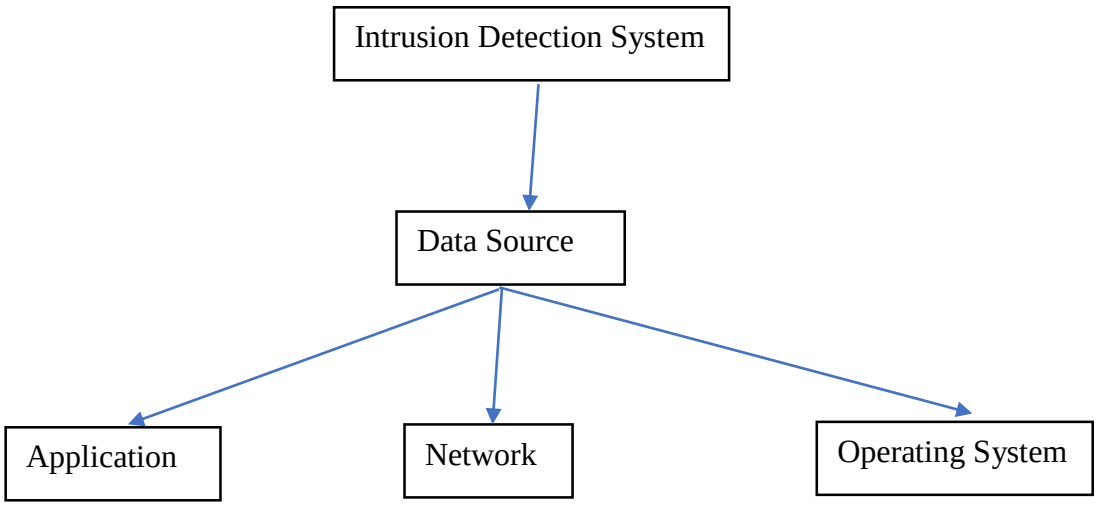


Figure 2: Various data sources in IDS

Application: Each application has a log maintained. IDS can be used to monitor these log files periodically. These types of IDS are named as application Intrusion detection system. They examine the operation executed in an application to ascertain if the application is being manipulated by other means that is prohibited. Example of such IDS are Discovery, RIPPER.

Network: In Network, communication data is analysed to determine if an attacker is trying to access other's network illegally. Examples such as EMERALD, BRO, SNORT and so on.

Operating System: The logs in this layer contain information from the kernel and other operating system components and help to determine if an attacker is trying to compromise the Operating system.

There are many example tools such as IDES, NIDES, USTAT.

In general, one of the best ways to restrict intruders is monitoring / auditing / analysing logs in the network environment. In each and every desktop computer we have antivirus, in companies we have multiple firewalls, and today even simple end-users buying the latest security tools. Among the above, how many are watching or monitoring all the information's which these tools generate. We have proxy logs, web server logs, and other authentication logs. A lot of attacks would not have happened if administrators cared to monitor their logs. In many networking environments, there is not even a network administrator. If there is, they are not able to

- Understand the logs;
- correlate the events (good or bad).

We try to make an attempt to establish a secured networking environment by this study for those who have not even got basic network knowledge.

2.6.4 Intrusion Detection Tools

There are a variety of tools and many are open source and only a few exist on most operating systems. Some of the more popular tools may need to be downloaded and installed. The common tools perform functions, and the specific tools test explicit functions. Query tools test connectivity to a particular network protocol. We can also term it as Network scanning tools.

The common tools are

2.6.4.1 Arp: A tool that list the local ARP table (Data link layer), modifies the ARP table, and more importantly, generates an ARP network request. If the arp command cannot identify the host on the local network, then the host is not on the local network.

2.6.4.2 Ping: A simple tool that generates an ICMP echo request. A successful ping indicates connectivity along the physical, data link and network layers.

2.6.4.3 Netstat: The netstat command is available on nearly all networked operating systems, although the parameters and output format may vary. This tool provides a variety of network related status reports, including current network connections, routing tables, and number of bytes transmitted and received.

Metrics can be displayed by network interface or protocol.

2.6.4.4 Telnet: Although originally designed for establishing a remote shell connection, telnet is commonly used to test connections to remote TCP services. Telnetting to a specific port on a remote host can test the connectivity of the entire remote network stack. The netcat application (nc) is a common alternative to telnet.

2.6.4.5 Nmap: Nmap[W8] is an extremely powerful tool for identifying remote hosts. Nmap is capable of identifying active hosts(e.g., an ICMP ping scan) and transport layer ports (TCP and UDP). In addition, it contains finger printing capabilities. Nmap not only identifies available TCP network services on a remote host, but it can attempt to identify the type of services.

2.6.4.6 Nessus: Nessus is a vulnerability assessment package that can perform many automated tests against a target network, including ICMP, TCP, and UDP scanning, testing of specific network services(such as Apache, MySql, Oracle, Microsoft IIs, and many others), and rich reporting of vulnerabilities identified.

2.6.4.7 Tracert: It is a command which can show the path a packet of information takes from a computer to one we specify. It will list all the routers it passes through until it reaches its destination, or fails to and is discarded. In addition to this, it will tell us how long each 'hop' from router to router takes.

2.6.4.8 RealSecure – RealSecure uses one or more PC based sensor engines which are essentially packed sniffers with packet filtering and collection rules and some alarm logic The first line of defense for any organizational network is typically a firewall and the second line is IDS. Contemporary IDSs look at each individual packet as it enters the network and based on information contained in packet headers tries to determine if the packet represents suspicious activity or not. Though there are a lot of IDS and each of them differ from others by classifying them based on their detection techniques almost all of them check both the payload and header information of the packets. For example, SNORT, Network IDS[W11] itself does it. When looking into the Free Open Source Software systems, it is found that SNORT and BRO[W12] are the two NIDS still active and research is going on. Both IDS perform Packet header analysis as well as Full packet analysis. In case of Host Based IDS OSSEC and Tripwire are Open Source.

It does not mean that all the IDS have to check both header and payload of each packet to find the attacks. Today, the payload may be encrypted in the IPv6 protocols. Internet Protocol Security (IPSec) is an Open source protocol that secures communication across IP based networks such as LANs, WANs and the Internet in Windows. The protocol is designed to protect data integrity, user confidentiality and authenticity at IP packet level. IPSec is the cryptographic authentication and encryption product of the IETF's IP Protocol Security working group. It is to be implemented in IPv6. It is retrofitted with IPv4. But due to lack of suitable standards, IPSec is not appropriate for some types of connectivity.

We are particular with attackers who want to:

- prevent access/use of systems and services (denial-of-service);
- scanning of the IP address and ports;
- compromise those systems which found (exploits).

The above attacks can be detected well by looking at TCP/IP packet header.
And more over, full TCP/IP packet header analysis will be CPU intensive.

2.6.4.TYPES OF IDS

Several types of IDS technologies exist due to the variance of network configurations. Each type has advantages and disadvantage in detection, configuration, and cost. Mainly, there are three important distinct families of IDS: The types of IDPS technologies are differentiated primarily by the types of events that they monitor and the ways in which they are deployed.

a) Network-Based

A Network Intrusion Detection System (NIDS) is one common type of IDS that analyses network traffic at all layers of the Open Systems Interconnection (OSI) model and makes decisions about the purpose of the traffic, analysing for suspicious activity. A term becoming more widely used by vendors is "Wireless Intrusion Prevention System" (WIPS) to describe a network device that monitors and analyses the wireless radio spectrum in a network for intrusions and performs countermeasures which monitors network traffic for particular network segments or devices and analyses the network and application protocol activity to identify suspicious activity. It is most commonly deployed at a boundary between networks, such as in proximity to border firewalls or routers, virtual private network (VPN) servers, remote access servers, and wireless networks. The NIDS are also called passive IDS since this kind of systems inform the administrator system that an attack has or had taken place, and it takes the adequate measures to assure the security of the system.

b) The Host Based IDS

According to the source of the data to examine, the Host Based Intrusion Detection System can be classified in two categories:

i) The HIDS Based Application.

The IDS of this type receive the data in application, for example, the logs files generated by the management software of the database, the server web or the firewalls. The vulnerability of this technique lies in the layer application.

ii) The HIDS Based Host.

The IDS of this type receive the information of the activity of the supervised system. This information is sometimes in the form of audit traces of the operating system. It can also include the logs system of other logs generated by the processes of the operating system and the contents of the object system not reflected in the standard audit of the operating system and the mechanisms of logging. These types of IDS can also use the results returned by another IDS of the Based Application type.

Host-based intrusion detection systems (HIDS) analyze network traffic and system-specific settings such as software calls, local security policy, local log audits, and more. A HIDS must be installed on each machine and requires configuration specific to that operating system and software. Host-Based, which monitors the characteristics of a single host and the events occurring within that host for suspicious activity. Host-based IDPSs are most commonly deployed on critical hosts such as publicly accessible servers and servers containing sensitive information.

Network Behavior Anomaly Detection

Network behavior anomaly detection (NBAD) views traffic on network segments to determine if anomalies exist in the amount or type of traffic. Segments that usually see very little traffic or segments that see only a particular type of traffic may transform the amount or type of traffic if an unwanted event occurs. NBAD requires several sensors to create a good snapshot of a network and requires benchmarking and baselining to determine the nominal amount of a segment's traffic. The NIDS-HIDS combination or the so called hybrid gathers the features of several different IDS. It allows, in only one single tool, to supervise the network and the terminals. The probes are placed in strategic points, and act like NIDS and/or HIDS according to their sites. All these probes carry up the alerts then to a machine which centralize them all, and aggregate the information of multiple origins.

2.7 DETECTION TYPES

2.7.1 Signature-Based Detection

An IDS can use signature-based detection, relying on known traffic data to analyze potentially unwanted traffic. This type of detection is very fast and easy to configure. However, an attacker can slightly modify an attack to render it undetectable by a signature-based IDS. Still, signature-based detection, although limited in its detection capability, can be very accurate.

2.7.2 Anomaly-Based Detection

An IDS that looks at network traffic and detects data that is incorrect, not valid, or generally abnormal is called anomaly-based detection. This method is useful for detecting unwanted traffic that is not specifically known. For instance, anomaly-based IDS will detect that an Internet protocol (IP) packet is malformed. It does not detect that it is malformed in a specific way, but indicates that it is anomalous.

2.7.3 Stateful Protocol Inspection

Stateful protocol inspection is similar to anomaly-based detection, but it can also analyze traffic at the network and transport layer and vender-specific traffic at the application layer, which anomaly-based detection cannot do.

False Positives and Negatives

It is impossible for an IDS to be perfect, primarily because network traffic is so complicated. The erroneous results in an IDS are divided into two types: false positives and false negatives. False positives occur when the IDS erroneously detect a problem with benign traffic. False negatives occur when unwanted traffic is undetected by the IDS. Both create problems for security administrators and may require that the system be calibrated. A greater number of false positives are generally more acceptable but can burden a security administrator with cumbersome amounts of data to sift through.

However, because it is undetected, false negatives do not afford a security administrator an opportunity to review the data. IDSs cannot provide completely accurate detection; they all generate false positives (incorrectly identifying benign activity as malicious) and false negatives (failing to identify malicious activity). which necessitates additional analysis resources to differentiate false positives from true malicious events. Most IDSs also offer features that compensate for the use of common evasion techniques, which modify the format or timing of malicious activity to alter its appearance but not its effect, to attempt to avoid detection by IDSs.

The primary classes of detection methodologies are as follows:

Signature-based

Which compares known threat signatures to observed events to identify incidents. This is very effective at detecting known threats but largely ineffective at detecting unknown threats and many variants on known threats. Signature-based detection cannot track and understand the state of complex communications, so it cannot detect most attacks that comprise multiple events.

Anomaly-based detection

Which compares definitions of what activity, is considered normal against observed events to identify significant deviations. This method uses profiles that are developed by monitoring the characteristics of typical activity over a period of time. The IDS then compare the characteristics of current activity to thresholds related to the profile. Anomaly-based detection methods can be very effective at detecting previously unknown threats.

Stateful protocol analysis

Which compares predetermined profiles of generally accepted definitions of benign protocol activity for each protocol state against observed events to identify deviations. Unlike anomaly-based detection, which uses host or network-specific profiles, stateful protocol analysis relies on vendor-developed universal profiles that specify how particular protocols should and should not be used. It is capable of understanding and tracking the state of protocols that have a notion of state, which allows it to detect

2.8 Issues and Challenges in IDS

- Deficiency or incomplete Data set.
- Detection Algorithms.
- Integration of multiple formats of data.

- Platform dependencies.
- Poor Design.
- Testing/ Evaluation of IDS.

Data Set

Data set can be defined as a collection of all the data or information during the survey which needs to be analyzed. Since in intrusion detection system, the data sets play important role in accuracy of results. Thus it became very much important to have datasets which are almost near to real time system. Now a days, the researchers are using data set DARPA 98, 99, New Mexico university immune system etc. but being outdated, we are not able to mitigate those attacks which are very much new. Therefore, this problem needs to be addressed in order to have most accurate and simplified results. Some popular data sets which are used by researchers for the purpose of experimentation but are outdated are as:

- MIT Lincoln laboratory -- DARPA intrusion detection and Evaluation.
- University of Mexico -- Computer immune system.
- University of California –UCI knowledge discovery in databases (KDD) Archive.
- University of Minnesota – MINDS
- Prude University –CERIAS Group.
- Naval Postgraduate school – intrusion Defense.
- University of Virginia – Application Intrusion Detection.
- University of California – State Transition Analysis Technique.

Detection policy

Detection policy –this is the main part while find whether the packet/ information come is attack or the useful information which the user needs to implement the process or jobs. The detection algorithm should be competent enough that it should match all the case in small time and also should match the terms efficiently. The detection policy may be either anomaly or mis-use based. In anomaly based detection, the behavior is identified and if behavior is identified as reverse of normal, it is declared as attack and in another scenario, the pattern is matched using some pattern matching algorithm for known attacks and if pattern matches fully with some suspicious data, it is declared as attack. But there are also drawbacks that there are no rules for new attacks to be matched, hence new attacks are not detected or if it makes some changes in data so that it cannot match the pattern, the attack is detected.

Integration of Multiple formats as we are well aware of the fact that the incoming frames or data may be in different formats. So, there is need that different formats shall be integrated on a single intrusion detection system i.e. on the fly it should check for the formats and check the stream for intrusions.

Platform Dependence

In current technological world, we have different / number of intrusion detection system available some are free source while other are commercial. While implementing these intrusion detection systems all of them have system requirement to implement the intrusion detection software. Therefore, needs some platform for implementation. As we do have different platforms, we need a intrusion detection software which may be platform independent so that we can implement the same intrusion detection software on all the platforms.

Poor design

The design of all the intrusion detection systems are compact *i.e.* if a user wants to change some part of the intrusion detection system, we have to stop the intrusion detection system, then made the changes as desired and re-deploy it again. Hence the design of the intrusion detection system must be like as mentioned below:

 It should have two parts, one core part which consists of detection algorithm and second part will be the part associated with pattern matching. This part should be updated on the fly. *i.e.* it should not affect the detection process of the system but only updates the other parts without touching core part of the system. Thus, every update should be added on the fly without stopping the intrusion detection system.

2.9 MACHINE LEARNING

Machine learning is a subfield of Computer Science. It is a type of Artificial Intelligence which allows programs to learn and find patterns within data. Machine learning explores algorithms that can learn from and make predictions on data. These algorithms are called machine learning algorithms. A machine learning algorithm has to learn before it can be used to make predictions on data. Learning means that the algorithm has to be shown several examples of data and what the correct predictions for these examples would be.

Once the machine learning algorithm has learned from the data, it can be used to make predictions on other data. During the learning phase, the machine learning algorithm is shown the heart rate of a patient and the current time.

There are two classes of machine learning algorithms. There is supervised learning and unsupervised learning. Supervised learning is trained using labeled data. Unsupervised learning uses unlabeled data. The data used to train machine learning algorithms is called a training set.

2.9.1 SUPERVISED MACHINE LEARNING

This method requires a labelled training set that contains both normal and anomalous samples for constructing the predictive model. Theoretically, supervised methods are believed to provide better detection rate than unsupervised methods. K-nearest neighbour(k-NN) is one of the most conventional nonparametric techniques that are used in supervised learning for anomaly detection. It calculates the approximate distances between different points on the input vectors and then assigns the unlabelled point to the class of its K-nearest neighbours. The Bayesian network is another popular model that can encode probabilistic relationships among

variables interest. This technique is generally used for anomaly detection in combination with statistical schemes.

2.9.2 UNSUPERVISED MACHINE LEARNING

These techniques do not require training data. They are based on two basic assumptions. First, they presume that most of the network connections are normal traffic and only a small amount of percentage is abnormal. Second, they anticipate that malicious traffic is statistically different from normal traffic. Based on these two assumptions, data groups of similar instances that appear frequently are assumed to be normal traffic and those data groups that are infrequent are considered to be malicious.

This can prove to be useful in areas such as banking security, natural sciences, medicine, and marketing, which are prone to malicious activities. With the machine, a learning organization can intensify search and increase effectiveness of their digital business initiatives.

2.9.3 EVALUATING ML FOR AN IDS

With a machine learning algorithm, performance can be measured using the F-score. However, for intrusion detection systems, this is not enough by itself. The F-score assumes that recall and precision have the same importance. This is not necessarily the case when evaluating intrusion detection systems. A false positive occurs when a sample is actually Normal but is classified as an Intrusion. A false negative occurs when a sample is actually an Intrusion but is classified as Normal. A false negative is bad, since it means that an Intrusion was not detected. But most IDS's are used in a layered approach. This means that if one layer does not detect an Intrusion, another layer might detect it.

Using ML for IDS

Data has to be processed before it can be used within a machine learning algorithm. This means that features have to be chosen. Some features can be easy to find, other have to found by experimenting and running tests. Using all the features of a dataset does not necessarily guarantee the best performances from the IDS.

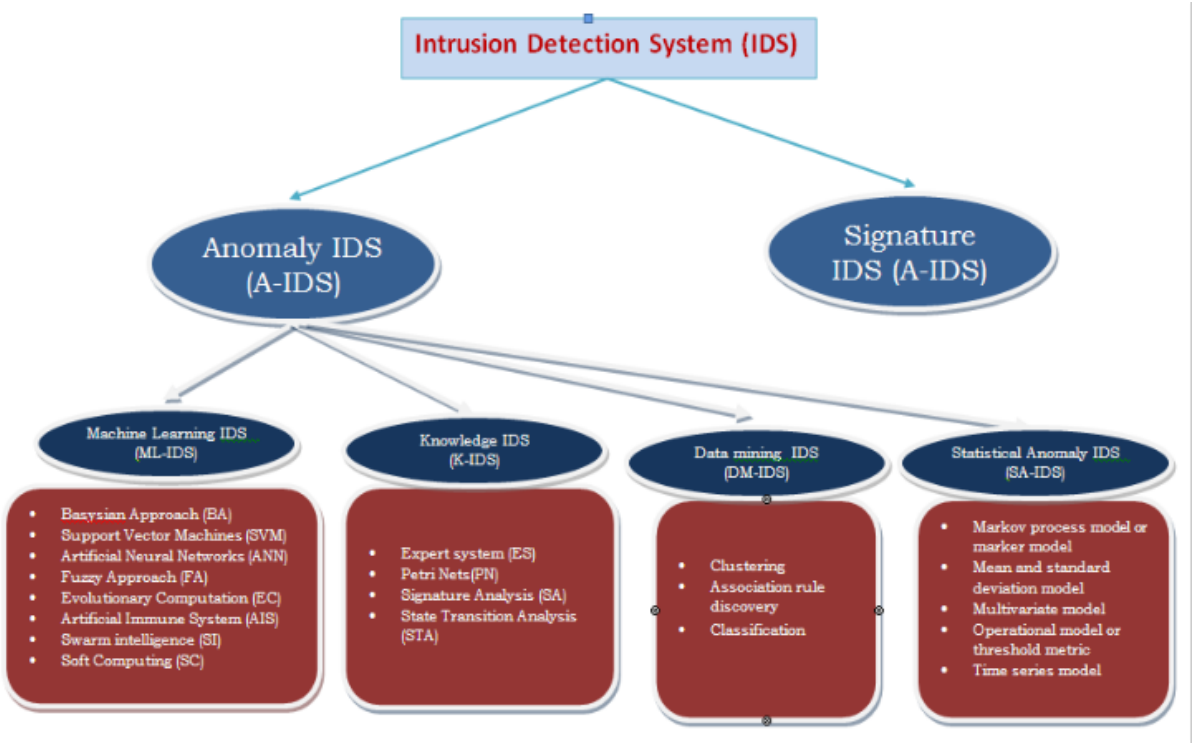


Figure 3: Types of Intrusion Detection System

Support Vector Machines (SVM): Vapnik 1998 was first to proposed Support vector machines. In this approach input vector is first mapped into the higher dimension feature space and there from SVM obtains the hyper -plans which are optimally separated in the higher dimensional feature space. The versatility and power of SVM comes from the fact that the decision boundary (separating hyper plan) is determine by support vector instead of training sample making it extremely regimental to outliers. Basically, SVM classifier works for binary classification i.e. separation of a set of training vectors belonging to different classes. Importantly in this approach support vector are same as training sample which are close to separating hyper plan.

Artificial Neural Networks (ANN): Based on a sequence of commands given by a specific user, a system using neural network approach learns to predict next command thus neural networks solve the problem of modelling user's behaviour in a continuous process which is used in anomaly detection because no explicit use model is needed. In expert system modelling researchers use d neural networks as an alternative to statistical technique for intrusion detection. It has been reported in the literature that the performance of both a basic signature matching system and a well-trained neural network were similar Goes *et al*.

• The user is to be identified by training the neural networks using command distribution vectors.

• Based on command distribution vector use neural network to check the user. An anomaly is detected in case the neural networks indicate a deferent user other than the actual user.

Knowledge Based Intrusion Detection System (K-IDS)

It is useful for signature-based IDS an also for detection of anomaly. Basically, it gathers the knowledge about system vulnerabilities and specific attacks. This knowledge is used to identify the attacks. If an event is not identified as attack no action is taken. This makes knowledge-based intrusion system relatively more accurate (less false alarms). The updating of knowledge of attack needs to be done regularly [20]. K-IDS classified into following categories

Signature Analysis (SA): Similar to expert system Signature analysis also uses knowledge - acquisition approach but in a different way. In this approach first attack description is converted in to semantic description which is again transformed i n to the information similar to audit trail. It has the advantage that it needs much less level of semantic description of attack making it efficient to implement. Therefore, it is mostly used in commercially available intrusion tools e.g. Haystack

COMPARISON OF VARIOUS INTRUSION DETECTION TECHNIQUES:
Comparisons are shown in table1.

Table 1: Comparison of various intrusion detection techniques

Approach	**Strengths**	**Weaknesses**
1. SA-IDS	1. Previous knowledge is not required. 2. accurate early warning is generated for long term attacks. 3. DoS attack are identified correctly. 4. Frequent Signature updates are not required. 5. Detects slow and low attacks. 6. Detects unusual activity. 7. It is capable of detecting the attack from part observation.	1. Correct statistical distribution is needed by statistical method give accurate result. However purely statistical method are not able to properly model all behaviours. 2. The basic assumption of quasi-stationary process is not fully met in real world intrusion detection system. 3. Even on relatively consistent network, SA-IDS learning process takes very long time to achieve accuracy and effectiveness.

		4. It is a tricky problem to set threshold to proper value. 5. Even for legitimate changes in user behaviour unacceptably high numbers of false alarms are generated.
2. DM-IDS	1. Researchers can focus on real intrusions because alarm data does not include data from normal activity. 2. "bad" sensor signatures and false alarm generators are identified. 3. Process narrows down to such anomalous activity that lead to a real intrusion. 4. Specially activities which continue for a long time are identified. (same activity ,different IP address)	1. Due to unpredictable changes in behaviour patterns of users a large number of false alarms are produced. 2. Even for characterizing normal behaviour of users, a large size of "training datasets" of system logs is often required.
3. K-IDS	1. The accuracy of the results produced by this technique is good. 2. False alarm rates is low. 3. It is flexible, scalable and robust. 4. Security officer can easily take preventive or corrective measures.	1. In order to keep this technique effective it is necessary that attack data is updated regularly. 2. Proper maintenance of knowledgebase is difficult because it requires proper analysis of each vulnerability. 3. Generalization is complex and time consuming task.
4. ML-IDS	1. Newly acquired information helps in Improving the performance by making appropriate execution strategy.	1. Resource expensive nature.

3 SYSTEM DEVELOPMENT

3.1 PROPOSED MODEL

A new intrusion model is being developed, where exiting algorithms will be suitably modified to make entire intrusion detection system adaptive and scalable.

The architecture of the model will support two data bases one storing known intrusion pattern and other storing regular usage pattern. The model will have the following characteristics:

1. Model will be adaptive in the sense that it will monitor the load on the server and frequency of intrusion so as to deploy one or multiple agents for pattern(s) matching.
2. Usage pattern will be first matched with the known intrusion pattern base and there after regular usage pattern database. This will make the detection faster.
3. Any new pattern will be temporarily stored and later on check it for being normal or intrusive and accordingly pattern databases will be updated after every fixed time window.

- Known algorithms will be suitably modified to for integration in the model. As an intermediary process systematic data mining approaches will be used to select the relevant system features to build better detection models.
- We propose to use (meta) learning agent-based architecture to combine multiple models, and to continuously update the detection models.

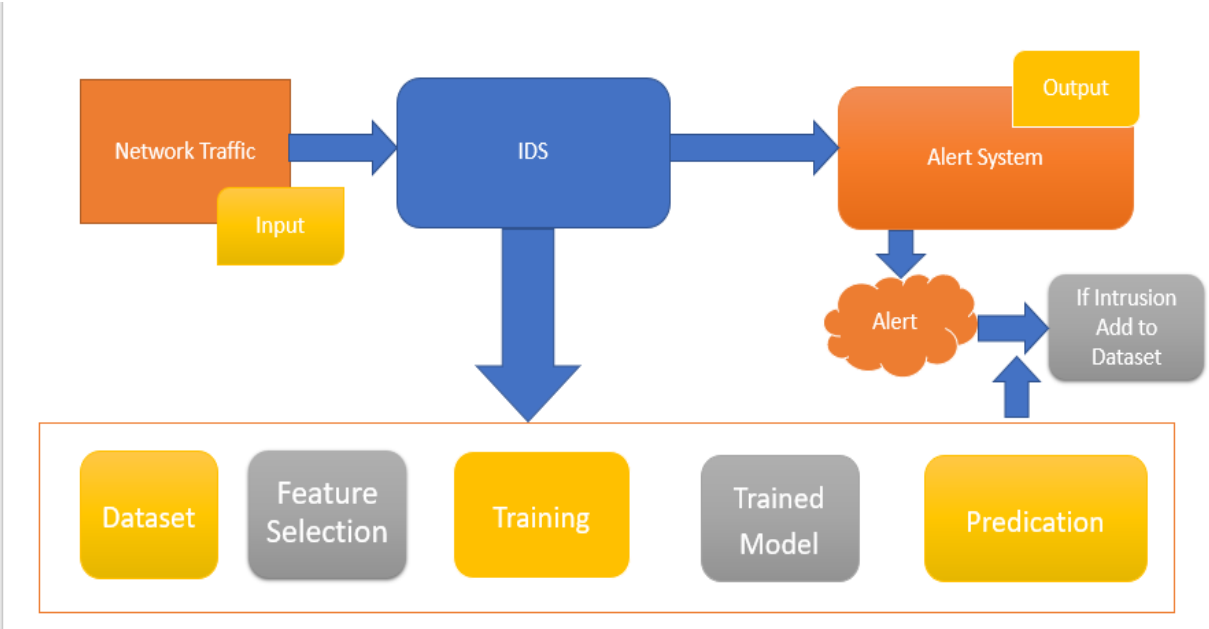


Figure 4. proposed model.

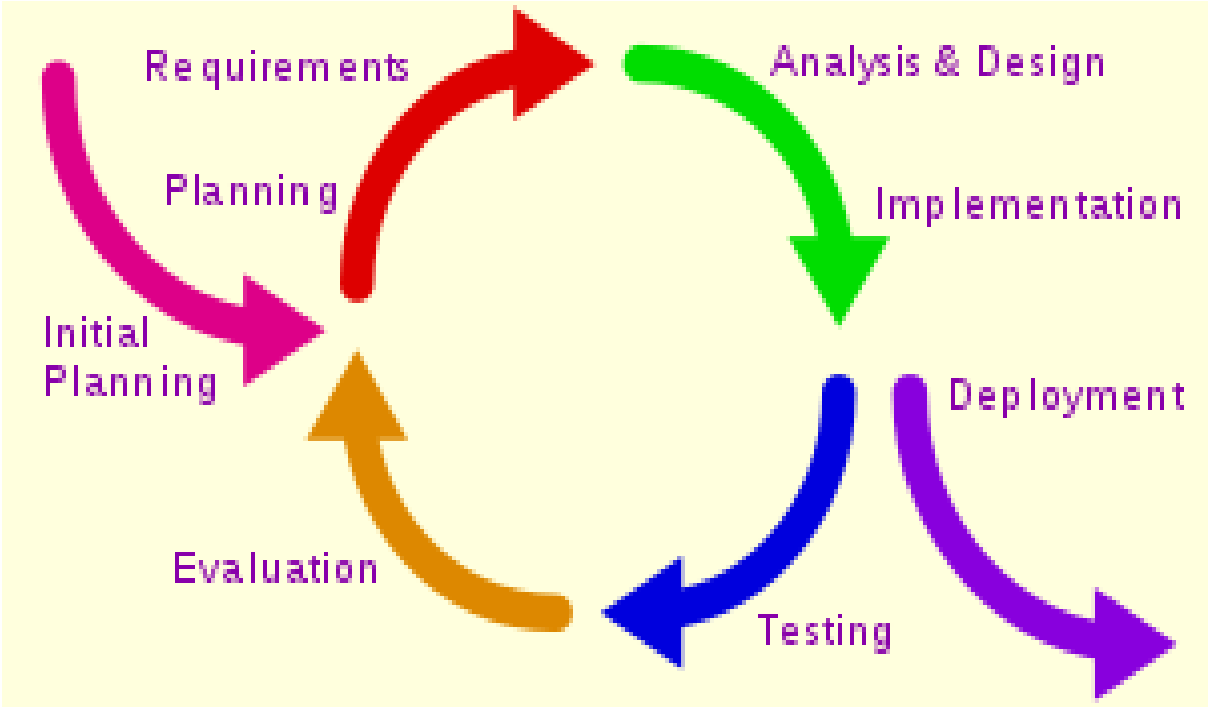


Figure 5. Iterative Development Model

3.1.1 Technology Stack

The main library that was used is scikit-learn. Scikit-learn is a robust machine learning library for Python. It is build upon NumPy, SciPy, and matplotlib. This library was chosen since the library offers the most important algorithms, the documentation. Scikit-learn also contain different methods to visualize machine learning algorithms such as a graph to show the learning curve. These can be a useful tool to evaluate the performance of machine learning algorithms. It also contains methods to calculate the F-score.

3.1.2 Program Execution

The implementation works in different steps. A JSON config file is used to define the elements that are used within the program. This contains the data to be used for learning, for checking, the machine learning algorithm, etc.. Once the config file has been read, the program can start the training phase. In this phase the specified algorithm is used and trained using the given data. Afterwards the prediction phase starts. This phase uses the prediction data and gathers all results. The structure of the program and the modules reflect these different phases.

3.1.3 Structure

The implementation is build to be modular. The first module is the machine learning module. This module contains all machine learning algorithms that can be used. There is also

a feature module. This module contains the available classes that can be used to extract features from the flows. A loader module contains all classes required to load the data from the different datasets.

A training module contains the different classes used for training. These classes use a loader class and pass the data to the machine learning algorithm. They define which data is supposed to be used (for example, using only abnormal behavior and leaving out the normal behavior). This results module receives all the output from the machine learning algorithms and has to log these or visualize them.

3.1.4 Datasets

In order to test the implementation and the algorithms, different datasets were used. Each dataset is used to test a different aspect of the machine learning algorithms. First, a subset of a dataset has to be chosen to be fed to the machine learning algorithms for learning.

3.1.5 Libpcap

In the field of computer network administration, pcap (packet capture) consists of an application programming interface (API) for capturing network traffic. Unix-like systems implement pcap in the Libpcap library; Windows uses a port of libpcap known as WinPcap. Monitoring software may use libpcap and/or WinPcap to capture packets traveling over a network. Libpcap and WinPcap also support saving captured packets to a file and reading files containing saved packets.

3.2 STATES OF MACHINE DEVELOPMENT

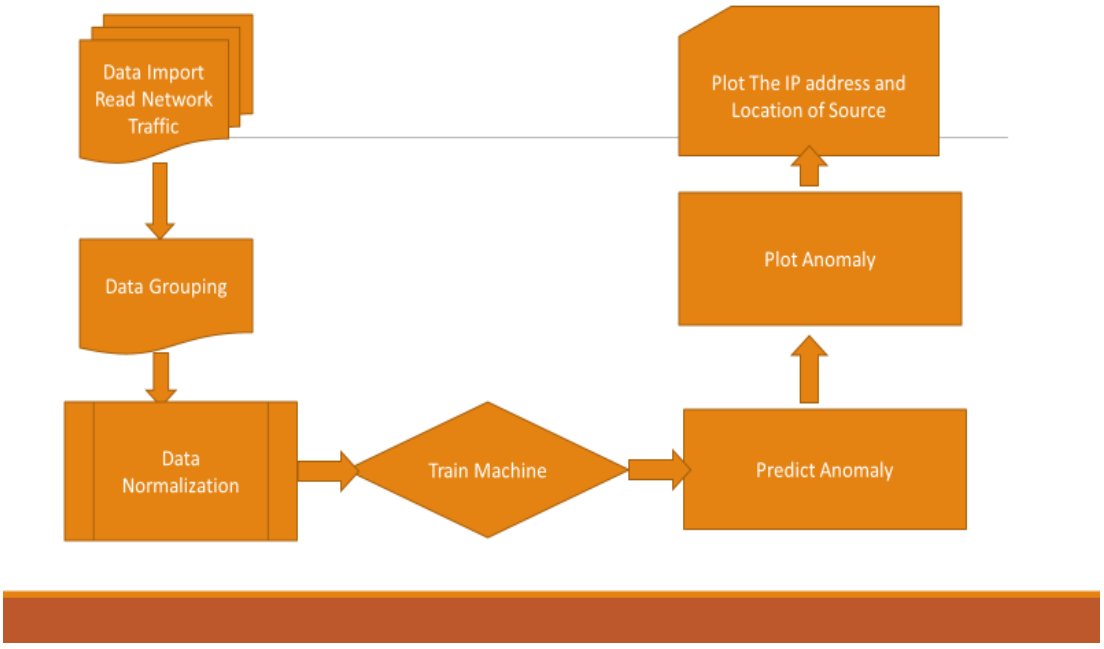


Figure 6: States of Machine Development.

Following figures are the actual work done on the IDS as proposed in the given above model:

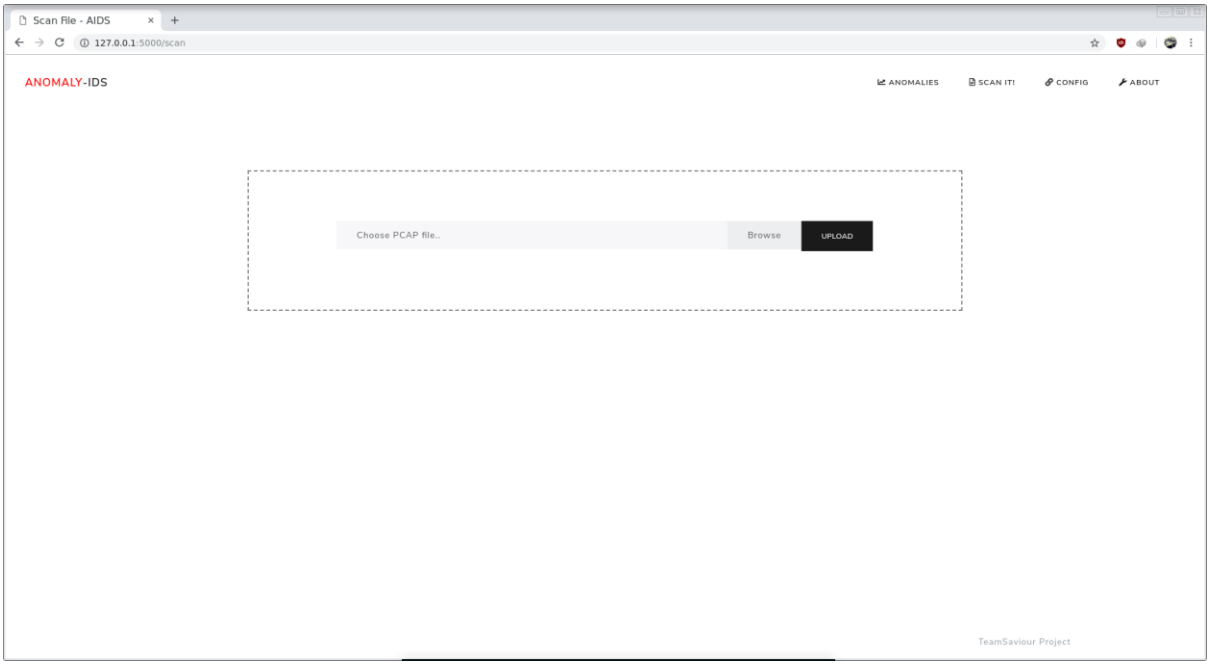


Figure 7. Selecting the dataset to scan it.

This process of selecting the dataset and processing it can be done live also, i.e. we can detect the live data packets coming to host or network by analyzing and sniffing the packets coming into the network like the packets can be monitored in the Wireshark here also we can monitor the live packets and detects if its an attack or regular packets by using isolation forest as algorithm to detect the anomaly by taking out outliers.

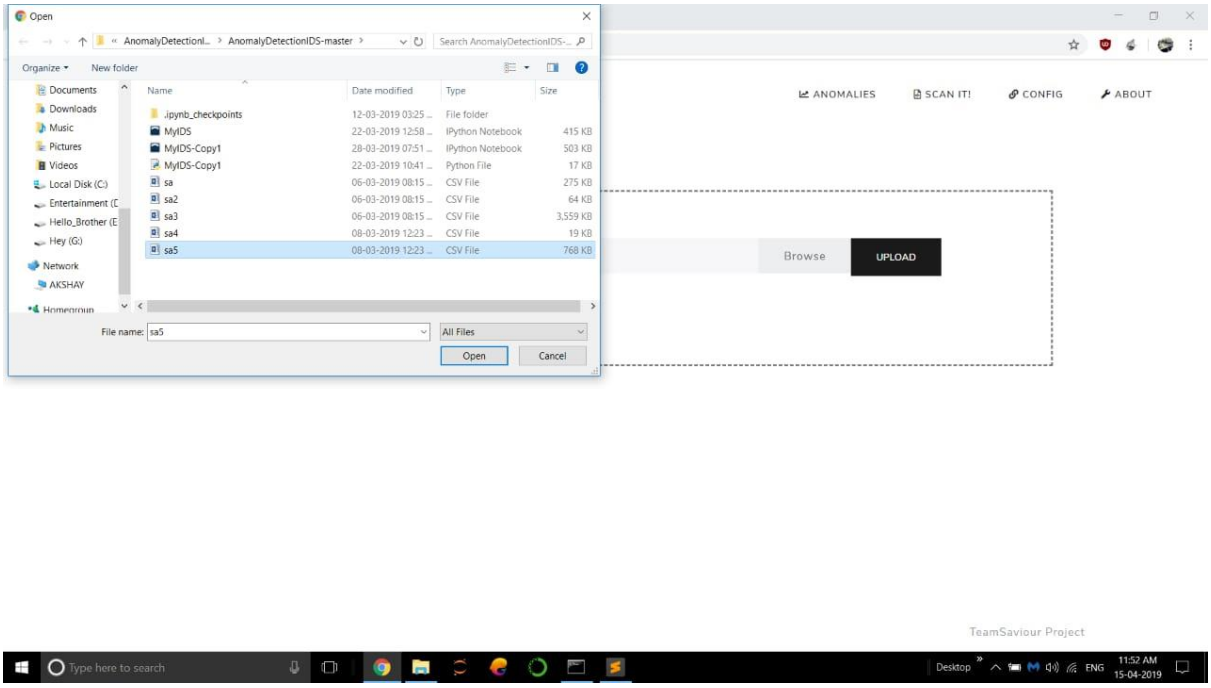


Figure 8. uploading the dataset to scan

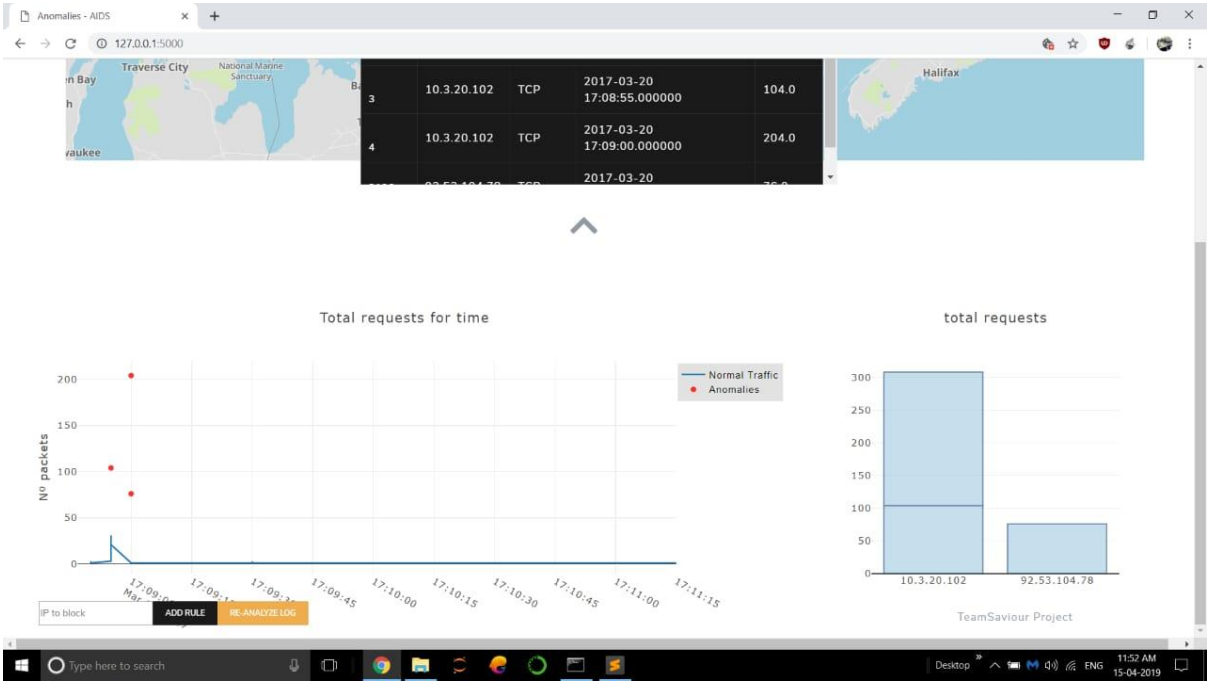


Figure 9. scanning the dataset.

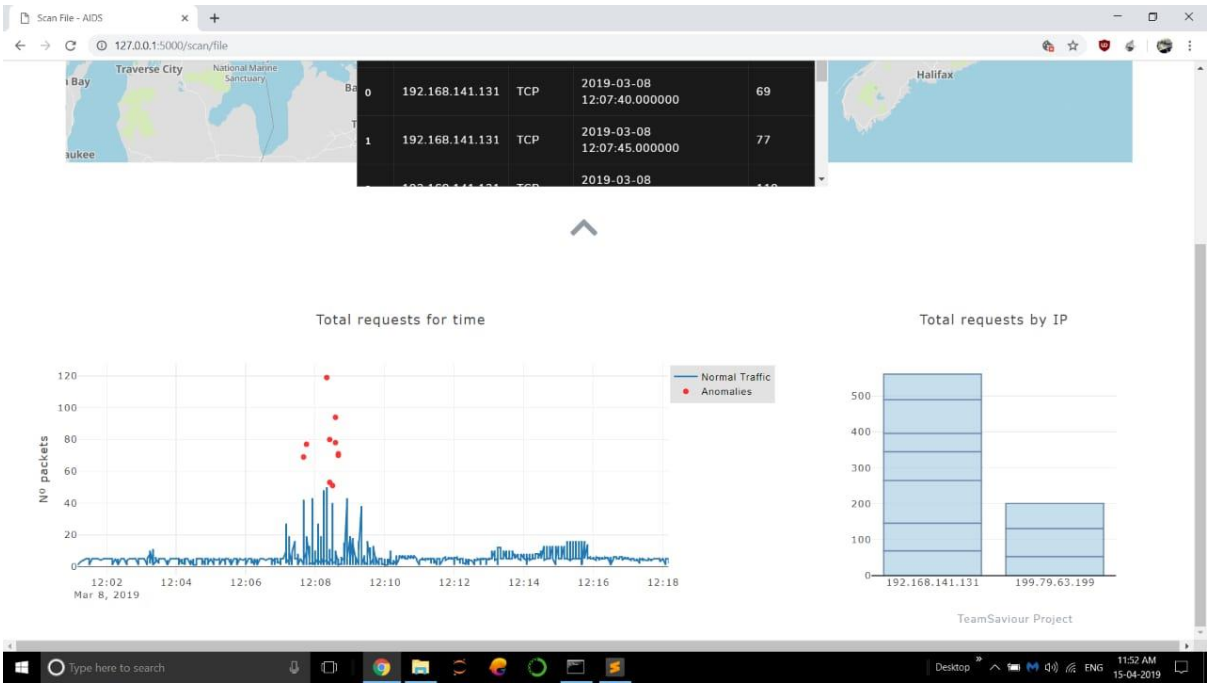


Figure 10. plotting the outlier's graph.

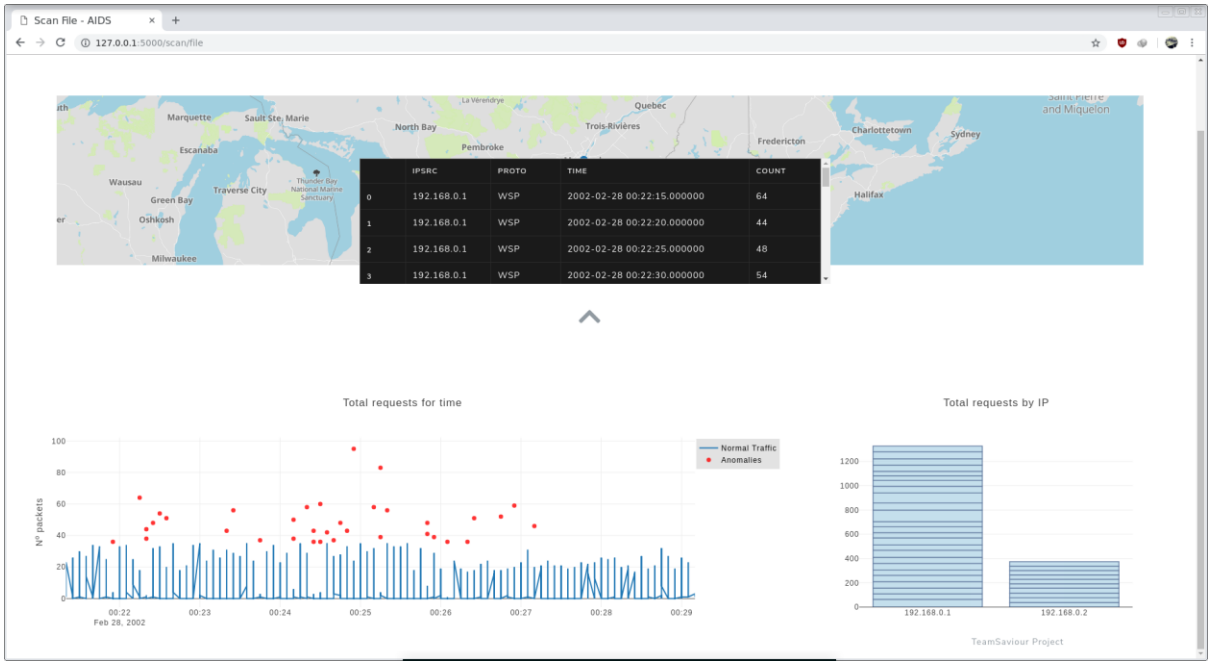


Figure 11. showing the ip addresses with anomalous behaviour.

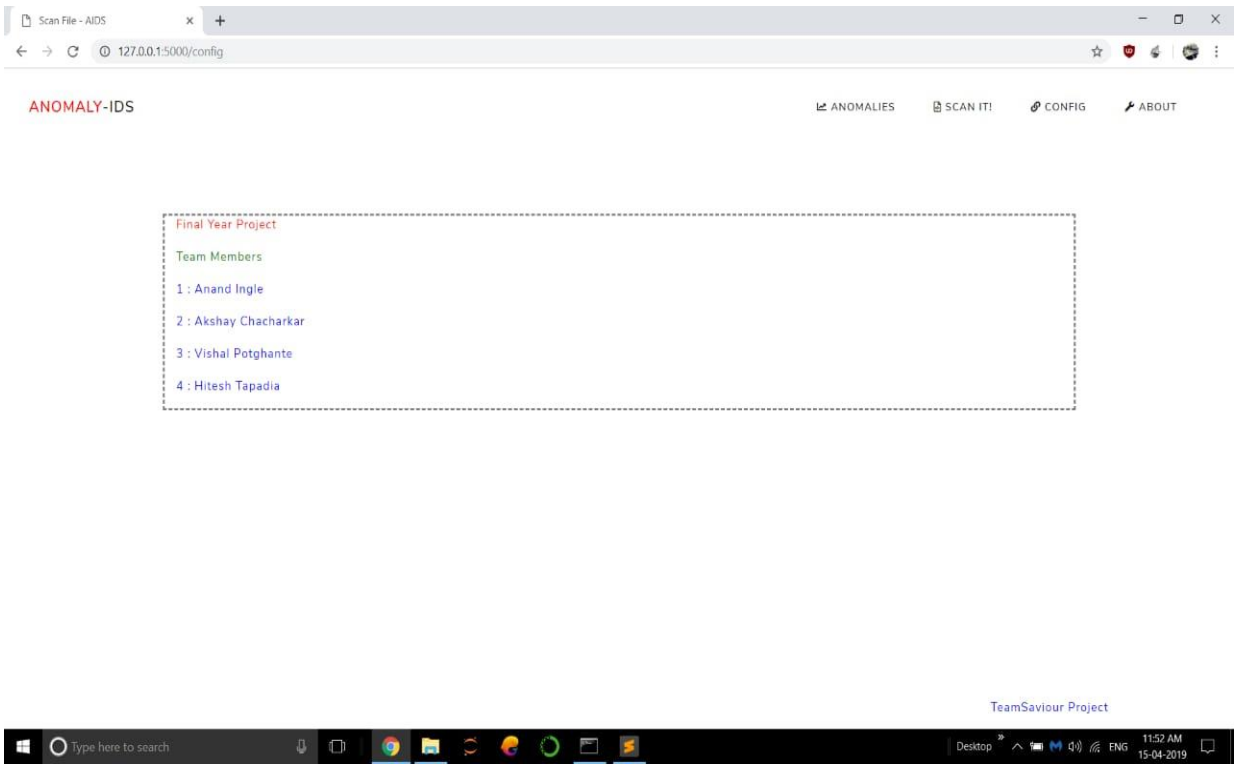


Figure 12. About the project developers.

3.3 Input Processing:

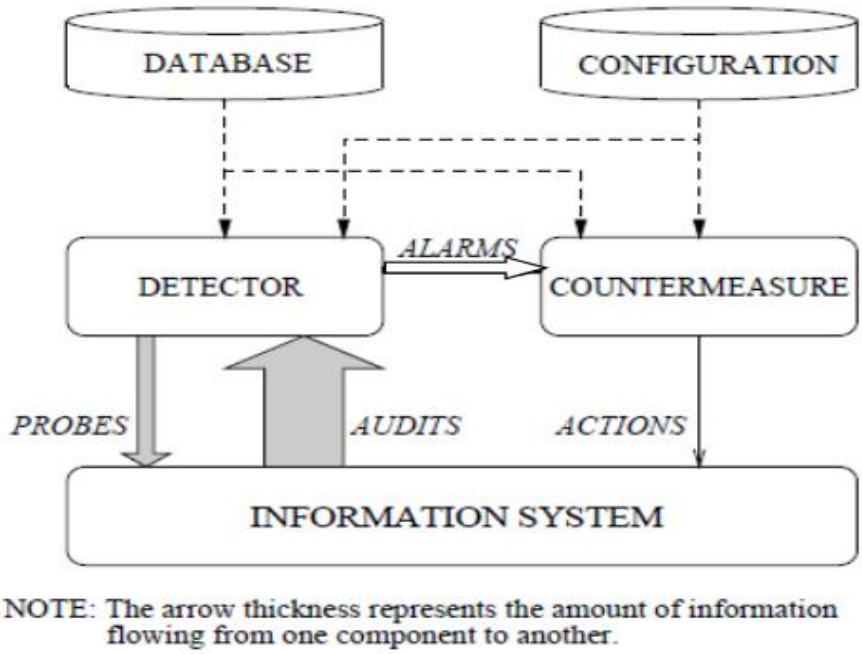


Figure 13. simple-typical IDS.

A common architecture for the structure of IDS comprises of a detection module which gathers data that contains evidence of intrusions, an analysis engine which processes the data for identification of intrusive activity and a response component which produces report for intrusions.

It is categorized on the basis of information being used by IDS. It is classified as,

Behavior Based: When the information is about the normal system behavior.

Knowledge Based: When the information is in relation to attacks.

The behavior on detection defines the response of IDS to attacks. The IDS is termed as,

- Active: When it reacts to the attack by taking either corrective or pro-active actions.
- Passive: If it functions only to generate the alarms.

The audit source location distinguishes the IDS on the basis of the type of input information analyzed. This input information can be,

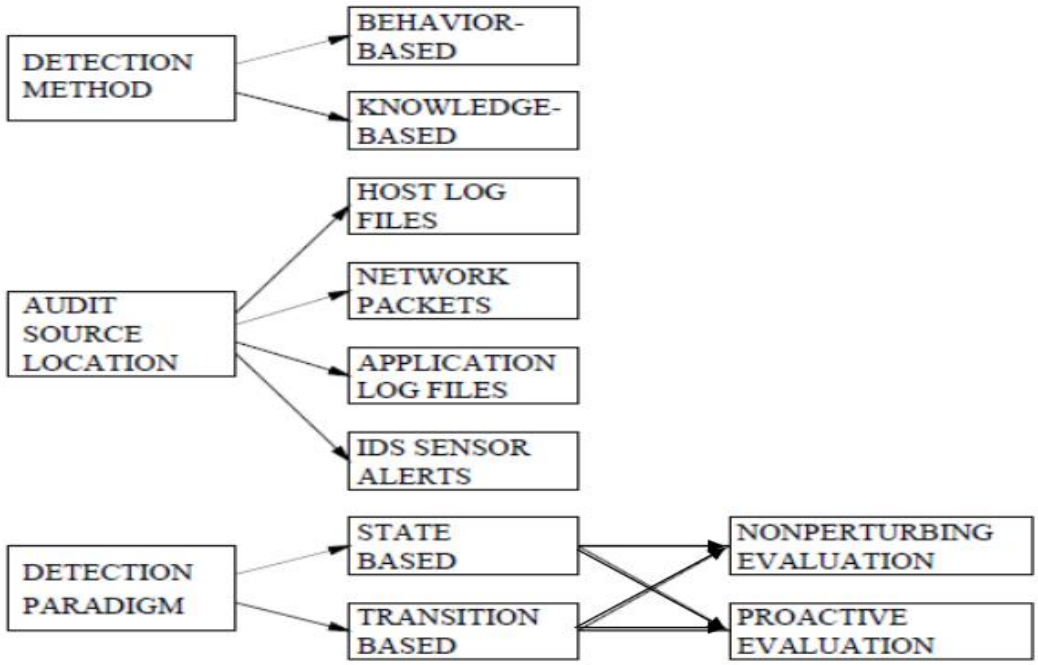
- System log files on a host
- Network packets
- Application logs

Intrusion detection alerts generated by other IDSs The detection paradigm illustrates the detection mechanism used by IDS. IDS can evaluate as,

- States

- Transitions

This evaluation can be performed in a non-obtrusive way or by actively stimulating the system to obtain a response.

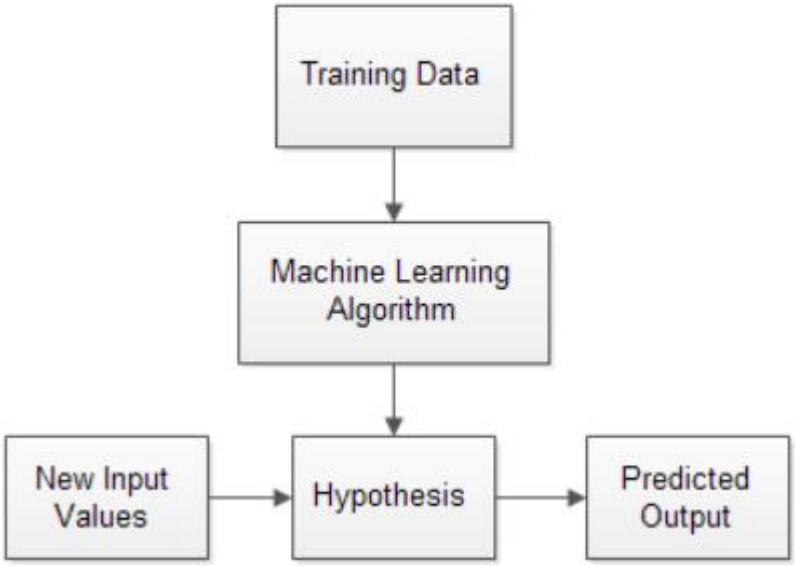


Attacks Commonly Detected By IDS's

Three types of attacks detected and reported by IDSs are,

- Scanning Attacks
- Denial of Service (DOS) Attacks
- Penetration Attacks
- 0 Day Attack

Figure 14: Flow of Machine Learning



The task of detecting network intrusions also come under machine learning as it involves the classification data into normal and abnormal behavior. Thus, for intrusion detection [87], we have,

•Task: To detect the intrusions in an accurate and precise manner.

•Experience: A dataset with instances representing normal and attack data.

•Performance Measure: Accuracy in the correct classification of intrusion events and normal events and other statistical metrics including precision, recall, F- measure and kappa statistic

Scikit-Learn Description

scikit-learn was used, which is a machine learning library written in python. Most of the learning algorithm implement in *scikit-learn* required data to be stored in a two-dimensional array or matrix. The size of the expected matrix is [samples, features].

The first parameter defines the number of samples, each sample is an item to be processed and the second parameter is the number of features that can be used to describe each item in a quantitative manner, generally real-valued but may be Boolean or discrete-valued in some cases. Data in *scikit-learn* is represented as a feature matrix and a label vector. Fig. 1 shows the data representation in scikit-learn.

$$\text{feature matrix}: \quad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ x_{31} & x_{32} & \cdots & x_{3D} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix}$$

$$\text{label vector}: \quad \mathbf{y} = [y_1, y_2, y_3, \cdots y_N]$$

TABLE 2. NSL KDD DATASET DESCRIPTION

Name of the Files	Description
KDDTrain+.TXT	It is the full training set including attack-type labels and difficulty level in csv format
KDDTest+.TXT	It is the full test set including attack-type labels and difficulty level in csv format
KDDTrain+_20Percent.TXT	20% subset of the KDDTrain+.txt
KDDTest-21.TXT	A subset of the KDDTest+.txt file which does not include records with difficulty level of 21 out of 21

Basic features, time-based traffic features, content features and host-based traffic features. All categories are described below:

Basic features: It contains all features which derived from TCP/IP connection such as Protocol type, Service, duration and etc.

- **Time-based traffic features:** It is used to capture those features which are mature over a 2 second temporal window (e.g. count, srv_count, Rerror_rate and etc.)
- **Content features**: Those features use domain knowledge to access the payload of the original TCP packets (e.g. hot, num_root, is_guest_login and etc.)

- **Host-based traffic features:** all attacks which span longer than 2 second intervals that have the same destination host as the current connection are access using these features (e.g. dst_host_count, dst_host_srv_count and etc.)

The classes or labels in the NSL KDD dataset are divided into four categories which represent the attack class and one as normal traffic:

1) **Denial of Service (DoS):** This attack aims to block or restrict a computer system or network resources or services.

2) **Probe**: here the intruder aims to scan for information or vulnerabilities in a network or computer system which later on will be used to launch attacks.

3) **Remote to Local (R2L):** Here the intruder gain remotely unauthorized access to a computer system over a network by sending data packet to that system.

4) **User to Root (U2R):** Here the intruder gains access to a user with normal privilege and later on try to access a user with administrator or root privilege.

The Table 2 and 3 describe and explain the analysis of the attack classes and types in the NSL_KDD dataset in details and shows the number of individual instances and records, both in the training and testing set.

Table 3 : Analysis of Attacks Classes and types

Attack Classes or Labels	Attack types (number of instances)	Total of instances
DoS	back (956), land(18), neptune(41,214), pod(201), smurf(2,646), teardrop(892)	45,927
Probe	satan(3,633), ipsweep(3,599), nmap(1,493), portsweep (2,931)	11,656
R2L	guess_passwd(53), ftp_write(8), imap(658), phf(4), multihop(7), warezmaster(20), warezclient(890), spy(2)	1,642
U2R	buffer_overflow(30), loadmodule(9), rootkit(10), perl(3)	52
Grand Total	59,277	

Table 4. Number of Instances In The Test Set

Attack class or label	Attack types (number of instances)	Total of instances
DoS	back(359), land(7), neptune(4,657), apache2(737), pod(41), smurf(665), teardrop(12), udpstorm(2), processtable(685), worm(2), mailbomb(293)	7,460
Probe	Satan(735), ipsweep(141), nmap(73), portsweep(157), mscan(996), saint(319)	2,421
R2L	guess_passwd(1,231), ftp_write(3), imap(307), xsnoop(4), phf(2), multihop(18), warezmaster(944), xlock(9), snmpguess(331), snmpgetattack(178), httptunnel(133), sendmail(14), named(17)	3,191
U2R	Buffer_overflow(20), loadmodule(2), xterm(13), rootkit(13), perl(2), sqlattack(2), ps(15)	67
Grand Total	13,139	

3.4 Data Cleaning and Pre-processing

Basically, in this step the dataset has to go through a cleaning process to remove duplicate records, as the NSL KDD dataset was employed which has already been cleaned, this step is not anymore required. Next a Pre-processing operation has to be taken in place because the dataset contains numerical and non-numerical instances. Generally the estimator (classifier) defines in the *scikit-learn* works well with numerical inputs, so a one-of-K or one-hot encoding method is used to make that transformation. This technique will transforms each categorical feature with *m* possible inputs to *n* binary features, with one active at the time only.

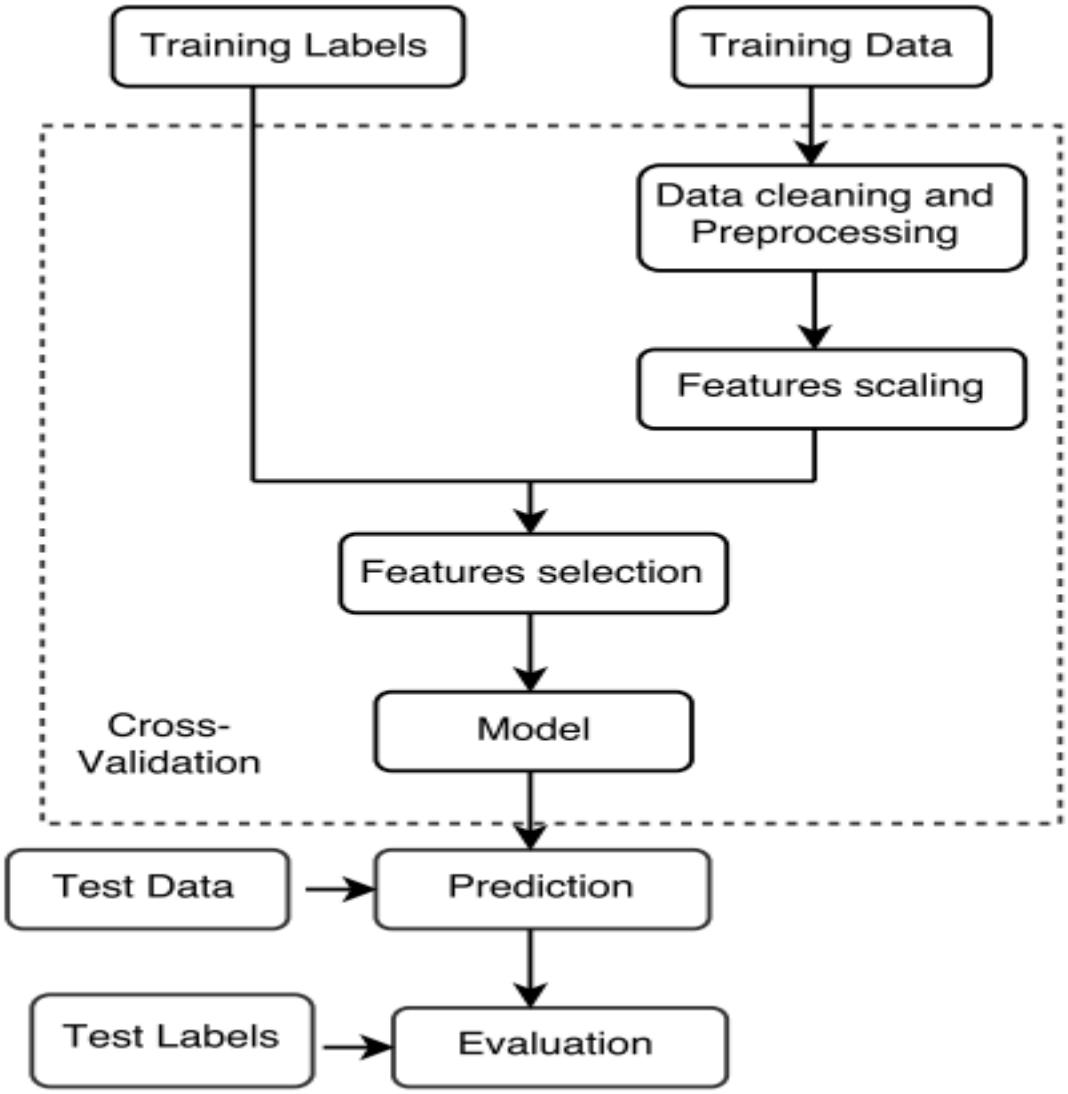


Figure 15: Training labels and data

Features scaling

Features scaling is a common requirement of machine learning methods, to avoid that features with large values may weight too much on the final results. For each feature, calculate the average, subtract the mean value from the feature value, and divide the result by their standard deviation. After scaling, each feature will have a zero average, with a standard deviation of one.

Features Selection

Feature selection is used to eliminate the redundant and irrelevant data. It is a technique of selecting a subset of relevant features that fully represents the given problem alongside a minimum deterioration of presentation.

Firstly, it is possible that irrelevant features could suggest correlations between features and target classes that arise just by chance and do not correctly model the problem. This aspect is also related to over-fitting, usually in a decision tree classifier.

The Best Features Subset Selection

The k-means classifier is used to compute the detection rate(accuracy) for each subset of features. Initially, the set of features S contains only the top ranked feature. When the accuracy drops, as an indication of model overfitting the algorithm is stopped.

Best Feature Selection Algorithm:

Input: F – Full feature set IGR: Information Gain Ratio Measure C: K-means classifier T: Gained Accuracy Threshold For each feature f compute IGR(f)

Output: S – Best feature subset

Algorithm

- Initialize**: S={}, ac=0** Repeat
- Assign acp= ac
- Evaluate f=getNext(F)
- Calculate S=SU{f}
- Calculate F=F-{f}
- Evaluate ac = accuracy(C,S)
- Continue the above steps until (ac-acp) < T Or ac <acp

3.5 Model

Here, a decision tree model was built to partition the data using information gain until instances in each leaf node have uniform class labels. This is a very simple but yet an effective hierarchical method for supervised learning (classification or regression) whereby the local space (region) is recognized in a sequence of repetitive splits in a reduced number of steps (small). At each test, a single feature is used to split the node according to the feature's values. If after the split, for every branch, all the instances selected belong to the similar class, the split is considered complete or pure. One of the possible methods to measure a good split is entropy or information gain. Entropy is an information-theoretic measure of the 'uncertainty' found in a training set, because of the existence of more than one possible classification. The training set entropy is represented by *H*. It is calculated in 'bits' of information and it is described as:

$$H = -\sum_{i=1}^{n} \mathrm{P}(c_i)\, log_2\, \mathrm{P}(c_i)$$

The generation process of a decision tree done by recursively splitting on features is equivalent to dividing the original training set into smaller sets recursively until the entropy of every one of these subsets is zero (*i.e,* everyone will have instances from a single class target).

A Decision Tree is made up internal decision nodes and terminal leaves. A test function is implemented by each decision node with a discrete result labelling the branches. Providing an input, at every node, a test is constructed and based on the outcome, one of the branches will be considered. Here the learning algorithm starts at the root and until a leaf node is reached, the process will be done recursively at which moment the value represented in the leaf node is the output. A leaf node can describe a localized space or region where instances finding in this input space (region) possess the same labels for classification and similar numeric value for regression.

4 PERFORMANCE ANALYSIS

4.1 Introduction to Testing and evaluation of IDS

Data is growing enormously and IDS has now become a standard for securing large network. Companies are investing huge amount in IDS technologies, but there is no such scientific methods to test the effectiveness of these IDS. Even though some quantitative measurable methods have been designed to test the effectiveness, but they do not evaluate the effectiveness on same scale. These methods consider coverage or probability of false alarm or probability of detection or resistance to attacks directed at IDS or ability of handling bandwidth and traffic or ability to identify attacks etc. Hence are not sufficient enough to figure out effectiveness of IDS. Also, there should be common scale for evaluating or testing the effectiveness of IDS. The different issues are as:

- Collecting script and victim software's.
- Different requirements for testing different types of IDS.
- Testing with different parameters.

4.2 Existing Tools and Methodologies

- DARPA and LARIAT [Environments]
- TCPReplay, IDSWakeup, WebAvalanche, HPING2 etc. [Tools]

Issues in developing such environment

1) Background Traffic
2) Database for attacks
3) Testing limited by case-by-case scenarios
4) High Costs and Security problems

Facts:

a. One test-bed for one set of related attacks.
b. IDS affected by system conditions – Stress.

4.3 Testing Methodology

4.3.1 General Methodology:

a. Create and select test scripts [normal/intrusion scripts]
b. Establish desired conditions – perf. Objectives.
c. Start IDS
d. Run Test Scripts
e. Analyse the IDS's output

Conditions

a. Intrusion Identification – Basic IDS test
b. Resource Usage – how much resources used by IDS.

Stress

a. Load – Testing IDS as low CPU priority task.[nice]
b. Intensity- Lot of activities generated in short time.
c. Background Noise
    i. Always created by “NORMAL” users.
    ii. e.g. Telnet Sessions associated with IDS host.

4.3.2 DARPA approach

a. Government undertaking – private and secure
b. Generate background traffic interlaced with intrusions.
c. Traffic can be generated by...
    i. Collect real data and attack actual org.
    ii. Sanitize data and introduce attack in data itself
    iii. Synthesize non-sensitive traffic from scratch

This approach had many shortcomings:

a. No effort to detect false positives.
b. Data rates and variation with time never considered. [stress]
c. Attacks were evenly distributed.
d. Size of training data may be insufficient.
e. Yet, DARPA was major effort to build such generalized Evaluation Environment for IDS testing.

4.3.3 LARIAT (Lincoln Adaptable Real-Time Information Assurance Test-Bed)

a. Emulates the Network Traffic from a small organization connected to Internet.
b. This was another attempt to build evaluation methodology.
c. Features:
    i. High Throughput capabilities.
    ii. Various attack scenarios
    iii. Windows Traffic in to account.
    iv. More Realistic and fully Automated

4.4 Tools For Testing IDS

1. TCPReplay: Provides background traffic by replaying pre-recorded traffic from network links.
2. IDSWakeup: Generates false attacks, in order to determine if IDS produces alerts.
3. WebAvalanche: Stress-Testing appliance for web applications and servers.
4. HPING2: Command line packet assembler and analyser.
5. Fragrouter: Routes network traffic such that it elude most NIDS.

4.5 Issues in Testing of an IDS

1. Traffic generation
    i. Background Traffic: contains non-malicious data.
    ii. Attack traffic: actual testing data for IDSs.
2. Databases
    i. Attacks intensity can vary in real-time
    ii. Databases need to be maintained and updated.
    iii. High cost
3. Effects of networking elements – Security Issue
    i. Firewalls, proxy server, ACLs etc.

4.6 Present Evaluation Environments

4.6.1 DARPA – Environment

i. Attack injection programs used to place attacks.
ii. Traffic generation was similar to early effort.
iii. Victim computer was anonymous FTP server.
iv. Environment focused on DOS attack**.**

4.6.2 Vendor Independent Testing Lab

v. Created by NSS group
vi. Build specialized lab to perform attacks on IDS
vii. Provides reports conversing large range of attacks.
viii. Focuses on user-interface, forensics and log management**.**

- Evaluation Environment – NOT just a Tool.
- No single methodology for testing IDS for every Attack.
- The BEST way: Evaluate IDS using live or recorded real – site specific traffic.
- DARPA experiment was significant
- Provides realistic evaluation environment
- Require lot of rework and not generalized**.**
- Development of IDS testing Methodology is in process.
- General, open-source and realistic Evaluation Environment is needed – NOT just a tool.
- Unless general methodology developed, IDS design and implementation will face problems.
- False positive and Misses
- Failure in Stress Conditions.
- IDS – Only a Part of Security!!

4.7 Using the Test Results

The test results can be used by the developers, users, and potential customers of an IDS to make the IDS more effective or to make a site more secure. A developer can use the results to find and correct weaknesses in the IDS. For example, if the tests show that the IDS is unable to detect a particular attack, the developer might enhance the language for describing attack signatures, so that the IDS could recognize that attack. Or, if the tests indicate that the IDS is consuming a large amount of resources (e.g., disk space), the developer might create a more efficient implementation that uses less resources. If nothing else, the developer might advertise the weaknesses revealed by the tests, so that users of the IDS can protect their sites by supplementing the IDS with other security tools.

An IDS user (e.g., a system administrator) may employ the test results to identify configuration problems, which may occur when the IDS has many configuration options or when the configuration steps are complex. If instead the user detects problems with the IDS itself, then the user can seek additional tools to protect the computer system. Finally, a potential IDS customer can use the test results to compare IDSs and thereby select the one that will perform best in the customer's computing environment.

5 CONCLUSION

5.1 conclusion

This thesis has given an overview of machine learning algorithms and has shown how they can be used in an intrusion detection system. Not all machine learning algorithms work as good. It is necessary to have a good labeled datasets which could be used to train the machine learning algorithms. If a good training dataset is used to train a machine learning algorithm, it can be used to create an intrusion detection system which offers acceptable performance out-of-the-box.

A lot depends on the quality of the training dataset. If the training dataset does not contain enough samples of the different intrusions, the machine learning algorithm will exhibit a large number of false positives and false negatives. Unsupervised learning algorithms do not work well out-of-the-box. They need a lot of manual interference before they are viable to be used for intrusion detection.

5.2 Future Work

The study performed in this project was a proof-of-concept. Therefore, several future improvements related to the practical implementation of this project can be identified:

5.2.1 Implement feature extraction in the inline mode

Currently, the feature extraction is performed after the files were run in the sandbox and the reports were generated. This approach will result in delays in the file analysis when implemented. In stead, it is advised to extract the features as they are processed by the sandbox, so that there will be no need to go through the reports again.

5.2.2 Use a wider dataset

Although the dataset that was used in this study is broad, covering most of the malware types that are relevant to the modern world, it does not cover all possible types. For more accurate evaluation of the predictors, it is advised to test the models on all the possible types of malware: spyware, adware, rootkits, backdoor, banking malware, etc. In a real-world application, the maximum amount of possible families should be used before the launch of the project for real-world environments.

5.2.3 Use pre-selected API's

In this work, the big overhead in the data processing was created by the need of selecting the relevant API calls and removing the redundant ones. For further implementation, only the APIs that were identified as relevant in this study can be used. This will decrease the amount of time required for data pre-processing, reduce the performance requirements of the machine on which the analysis is being done and decrease the level of feature selection to be made. However, it should be noted, that for more accurate description, the relevant APIs should be extracted from the biggest possible dataset. Also, it is advised to select the relevant APIs per malware family, as this will result in another level of flexibility and accuracy.

REFERENCES

1. www.researchgate.net

2. www.secureworks.com

3. Intrusion Detection Based On Artificial Intelligence Technique –International Journal of Computer Science Trends and Technology (IJCST) – Volume 2 Issue 4, July-Aug 2014

4. Parsazad, S., Saboori, E. & Allahyar, " Fast Feature Reduction in intrusion detection datasets, " MIPRO 2012 Proceedings of the 35th International Convention, pp.1023–1029.

5. Scikit-Learn, Accessed December 2015, http://scikit-learn.org/stable/index.html

6. N.G. Relan and D. R. Patil, "Implementation of Network Intrusion Detection System using Variant of Decision Tree Algorithm," 2015 Int. Conf. Nascent Technol. Eng. F., pp. 3–7, 2015.

7. Dewan Md. Farid, Nouria Harbi, and Mohammad Zahidur Rahman, "Combining Nave Bayes and Decision Tree for Adaptive Intrusion Detection," International Journal of Network Security & Its Applications, Vol. 2, No. 2, April 2010, pp. 12-25.

8. Bhavsar Y. B, Waghmare K. C. "Intrusion Detection System Using Data Mining Technique: Support Vector Machine," International Journal of Emerging Technology and Advanced Engineering, Vol.3, Issue 3, pp.581-586(2013).

9. Intrusion Detection and Prevention System: Issues and Challenges Bilal Maqbool Beigh, Uzair Bashir, Manzoor Chachoo(IJCT 2014)

10. Intrusion Detection System Based on Multi-class SVM Hansung Lee, Jiyoung Song, and Daihee Park

11. Host-based IntrusionDetection Giovanni Vigna

12. ML-IDS: A machine learning approach to detect wormhole attacks by P Shukla - 2017

13. Pieta, Nicholas J.; Chung, Mandy;, Olsson, Ronald A and Mukherjee, Biswanath. "A methodology for testing Intrusion Detection Systems", IEEE Transactions on Software Engineering, 22, 1996, ppl. 719-720.

14. Athanasiades, Nicholas;Abler, Randal;Levine, John; Owen, Henry;Riley, George. "Intrusion Detection Testing and Benchmarking Methodologies", IEEE International Information Assurance Workshop, 2003

ACKNOWLEDGEMENT

We would like to express our deep sense of gratitude and appreciation to all those who gave me the opportunity to complete this project and report. A special thanks to our final year project guide Prof. S.M. Chavan, whose help, stimulating suggestions and encouragement helped us to coordinate our project especially in writing this report. We would like to thank the H.O.D. Prof. C. M. Gaikwad of Information Technology Department, of Government college of Engineering, Aurangabad for giving me such a chance to commence this work.

We would also like to acknowledge with much appreciation the crucial role of the staff of Information Technology Department, who supported us and gave their valuable time to clear our doubts regarding the project report.

A special thanks goes to our classmates for lending their support, pacing up our project and giving their valuable suggestions for betterment of our project.

Anand Ingle BE16S06F006
Akshay Chacharkar BE15F06F008
Hitesh Tapdiya BE15F06F054
Vishal Potgante BE15F06F043