JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

Modelling an Intrusion Detection system using ensemble approach based on voting to improve accuracy of base classifiers

[1]Dr.Anwaar Ahmad Wani, [2]Dr.Juneed Iqbal, [3]Dr.Mudassir Makhdoomi,

Assistant Professor (C), Islamia College of Science &Commerce, Srinagar,UT-J&K., India.
Email: anwaar1007@gmail.com
Assistant Professor, Islamia College of Science &Commerce, Srinagar,UT-J&K., India..
Email: juny@live.in
Assistant Professor, Islamia College of Science &Commerce, Srinagar,UT-J&K., India.
Email: mudassir76@rediffmail.com

ABSTRACT

Security is always a major concern when people put their money or assets online. Since a network is an open road for data, unwanted users always try to make illegal use of these networks. Many important measures like authentication, authorization and use of firewalls were used in early stages to prevent such illegal access into systems. This work is aimed to study the various techniques and algorithms of an IDS and pick out some good techniques by doing a comparative study and then try to improve and overcome the design parameters of these techniques such that the overall design and functionality of an IDS is enhanced. In our proposed empirical/theoretical model the dataset which was reduced to a set of most prominent features based on experiment on three parameters. Intrusions are defined as activities or events that violate the confidentiality, integrity and availability of a computing system. Advancements in the field of artificial intelligence has given a different face to IDSs. In our model we have used three base classifiers. Our descriptive and comparative study helped us to short list the three algorithms. We have also taken a technique from the theory of induction through decision trees which is considered to be an ensemble in itself. Hence the overall accuracy of random trees is already better. This has helped to improve the overall accuracy of the proposed system. The final proposed model turned out to be more accurate and the time taken was also reduced by a large extent.

Keywords: IDS, Ensemble learning, Classifiers, Attacks, Security.

1. Introduction

Network security continues to be an interesting field primarily because of increasing number of intrusions taking place every day. Apart from human origin (called intruders), security can be breached by natural disasters, catastrophic failures and many other sources. Threats, except from those that originate from people are not a major cause of concern. These however, can be dealt by keeping copies of data at different locations, keeping backup generators and so on. The intruders find one way or the other to breach into the computing system to compromise the security model of an organization. This puts a challenge in front of security engineers who continuously work to identify the loopholes and limitations of these security measures and try to improvise accordingly. The fact, a computing system is not only insecure from the external environment, identifying the intruders from within the system is in itself a difficult if not impossible task.

Security techniques such as authentication **[1]** and access control **[2]** have been developed to achieve security at some level - namely to prevent intruders from accessing and manipulating information. Security systems, like firewalls, have succeeded in some ways but they provide minimal security at internal level. These prevention systems act as first level of defense against simple intrusions. The advantages of the Internet, namely the availability and amount of information, also pose the largest threat to information security. Use of Intrusion Detection Systems provide us with second level security defense systems **[3].**

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

Intrusion detection systems (IDS) are the software or hardware implementations of actions that are meant to monitor the computing system against those activities or events that compromise the security of a system. The threats to the computing systems are result of actions by those persons who try to take advantage of the loopholes of a system. These loopholes may be the result of an improper implementation of a security model, weak programming techniques, illegal use of privileges, and so on. Moreover, even if these security mechanisms can protect information systems successfully, it is still desirable to know what intrusions have happened or are happening, so that we can understand the security threats and risks and thus be better prepared for future attacks. In spite of their importance, IDSs are not replacements for protective security mechanisms, such as access control and authentication. Indeed, IDSs themselves cannot provide sufficient protection for information systems. As an extreme example, if an attacker erases all the data in an information system, detecting the attacks cannot reduce the 12 damage at all.

1.1 Components of an IDS

An IDS is a very complex system when it comes to its structure. It has many components working in active and passive modes to continuously monitor and detect intrusions. The characteristic components of an Intrusion Detection System are as follows:

Sensor or Agent: Sensors or agents observe and scrutinize activity. Their job is to continuously monitor the traffic which flows through them and pass them along to management servers.
Management Server: The job of management server is to examine the data that it receives from multiple sensors or agents. The data is usually in raw from as the sensors are not capable of processing any information. There might be one centralized management server or multiple servers for multiple sensors. They identify the events or activities in the data and try to draw similar patterns. This process is called correlation. Depending upon the type of application, we can have both hardware based or software based servers.

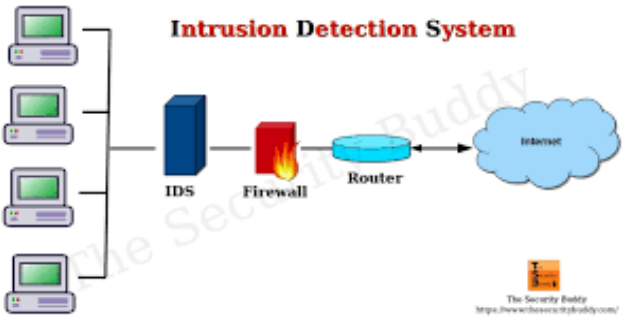

Figure 1.1: Structure of an IDS

Database servers: All the events that are recorded and processed are finally put up in a repository which can be used to deal with the intrusions that may arise in future. This also helps in analyzing similar patterns for new intrusions. Success of IDS particularly depends on how well it identifies the intrusions that occur over the time.

Console units: These are interfaces that help administrators to interact with a particular IDS. These are usually installed as separate systems (desktop or laptops) over the network. Some consoles are used for monitoring and inspecting the network traffic while others may be used to configure and update the systems over the time to improve the efficiency of intrusion detection systems.

2. Literature Review

Researchers have already invested some time and effort in studying and designing ensemble for Network Security in general, and Intrusion Detection Systems in particular. The work carried out suggests strong potential in using ensemble methods for the purpose of classification and thus can be used for both IDSs and IDPSs. In 2003, Mukkamala et al. **[4]** integrated Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs) into an ensemble based approach

and their empirical study suggested that the ensemble approach of these base algorithms outperforms both individual ones in terms of accuracy. The authors have used DARPA dataset to conduct the experiments. Chebrolu et al. **[5]** discussed the hybridization of Bayesian Networks with Classification and regression trees. The authors emphasize on feature selection/ reduction to design lightweight IDSs. The authors use KDD Cup 99 Intrusion Detection Dataset to make an empirical study of their approach. First, the authors make an empirical study of full and reduced datasets on individual algorithms and then they compare their results with the final ensemble approach. For the purpose of feature selection the authors have used Markov blanket & decision tree analysis. The results showed that the ensemble performed better with 100% accuracy on Normal, Probe and DoS attacks and 99.4% and 84% in R2L and U2R respectively. In 2005, Peddabachigari et al. **[6]** studied and presented ensemble approach based on Decision Trees and SVMs. They presented a hierarchical IDS based on these two classifiers. This IDS used four components to classify the dataset (KDD Cup 99 Dataset). First, it used decision trees to detect U2R attacks; Second, it used SVM to detect DOS; Third, it used a hybrid Decision Tree based SVM (DT-SVM) to classify Normal data; and finally an ensemble of the above three classifiers was used to detect attacks based on Probe and R2L. Zainal et al. **[7]** perform an empirical study forming a one Class classifier based on ensemble of Linear Genetic Programming (LGP), Adaptive Neural Fuzzy Inference System (ANFIS) and Random Forest (RF) for improved accuracies. The authors have also used KDD Cup 99 dataset for the purpose of evaluation and analysis. They have also applied feature reduction to the dataset before testing the ensemble. Experiments show the ensemble developed outperforms all the base classifiers in terms of accuracy and reduced false positive rates for a five class classifier. In the paper "Intrusion Detection in Computer Networks by a Modular Ensemble of One-Class Classifiers" **[8]**, authors discuss a modular Multiple Classifier System (MCS) where each module models a group of protocols or network services with similar characteristics. This models allows for use of different models and thresholds for different groups. Experimental results were conducted using KDD Cup 99 Dataset.

3. Analysis Techniques used by IDS

IDS either uses below mentioned techniques (either one of these or a combination of any of these) to safeguard a system:

1. **Code-Behavior Analysis:** aims at identifying malicious activity by analyzing attempts to execute code. For example, code-behavior analysis can first execute code in a virtual environment and compare its behavior to profiles or rules; buffer overflow technique works by finding scripts of code that cause the stack or buffer overflows. Continuous draining of buffer is presumed to be related to an intrusions or an abnormal behavior.

2. **Network Traffic Analysis & Filtering:** analyzes network, transport and application layer protocols. Sensors often include a host-based firewall that can restrict incoming and outgoing traffic for the system .

3. **File System Monitoring**: includes a number of methods, such as file-integrity checking, file-attribute checking; these two methods can only determine after-the-fact if the file has been changed. Some sensors use a small library and perform transparent analysis to monitor all attempts to access critical files and attempts that are suspicious. The current attempt is compared against a set of policies regarding file access and blocked if the type of access that has been requested (read-write-execute) contradicts a policy.

4. **Log Analysis**: Some sensors can identify malicious activity by monitoring and analyzing system and application logs which contain information e.g., shutting down the system, starting a service, application startup and shutdown, failures, configuration changes, etc.

5. **Network Configuration Monitoring**: Sensors are able to monitor a host's current network configuration and detect changes to it. For example, using ports like TCP and UDP for unusual activities or making use of unusually more number of ports, deactivating the network interface cards and similar situations marks that the network or the host on a network has been compromised.
6. **Process Status Monitoring**: Some IDSs can monitor the status of the processes and services running on a host. They restart by design when they detect that the host has stopped. This provides protection against some forms of malware which can sometimes disable antivirus and the similar applications.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

3.1 Types of Analysis Techniques

- **Computational Intelligence Approach:**

The voluminous data which is checked for patterns in intrusion detection systems were the first implementations of computational intelligence **[9].** Raw data is converted into ASCII network packet information, which in turn is converted into connection level information **[10**]. These information particularly relates to connection features like connection type, service, duration etc. Besides machine learning techniques, several intelligent paradigms have been explored to create models to detect intrusions.

- **Artificial Neural Network:**

These models are based on the structure of neurons in the human brain. Artificial Neural Networks (NNs) **[11**] can be trained to recognize arbitrary patterns in input data and associate such patterns with an outcome, which can be a binary indication of whether an intrusion has occurred [Suz06]. Such models are only as accurate as the data used to train them.

- **State Transition Tables:**

It is a finite automata based approach which generates a tables representing different states of transition in which a finite state machine will move to. State Transition Tables describe a sequence of actions an intruder does in the form of a state transition diagram. An intrusion will be flagged if any transition matches the states that are placed in those tables.

- **Genetic Algorithm:**

Genetic Algorithms **[12] [13**] - an evolutionary algorithm, mimics the natural reproduction system in nature where only the fittest individuals in a generation will be reproduced in subsequent generations, after undergoing recombination and random change. In mid 90's, the IDS were designed using the concept of GA's, and intrusions are detected by evolving signatures .A related technique is the Learning Classifier System (LCS), where binary rules are evolved, that collectively recognizes patterns of intrusion. Genetic Programming (GP) is a technique of computational intelligence used with other techniques like fuzzy logic, neural networks, that helps programs evolve. Its main characteristics in comparison to tree-based GP lies in that the evolved units are not the expressions of a functional programming language (like LISP), but the programs of an imperative language (like C/C++). In Multi Expression Programming (MEP), a chromosome encodes more than one problem solution. The expression that best matches the encoded chromosome defines the chromosome fitness in the whole solution.

- **Bayesian Network:**

Bayesian Network is a graphical representation of the joint probability distribution function over a set of variables **[14]**. In Bayesian Network, the network structure can be represented as a Directed Acyclic Graph where each node contains the state of random variable and a conditional probability table, which determine the probabilities of the node in a state, given a state of its parent **[15].** Individual events which occur during an attack are represented as nodes in the graph and relationship between those events are represented as edges of the graph and this graph is then used to detect the intrusion. It is best suited for Intrusion Detection systems because the models better represent the events that have occurred and tend to draw information as to which parameters and their respective values lead to an intrusions. They are also heavily used for drawing disease information from the symptoms known in advance. They tend to be more popular as they can draw results based on incomplete information.

- **Fuzzy Logic:**

Fuzzy Logic is a set of concepts and approaches designed to handle and manipulate elusive, noisy and imprecise data. It defines the percentage of a solution in a computing problem rather than directly stating the values of truth and false **[16].** A relationship is defined between the input variables and the output variables, which may indicate whether an intrusion has occurred.

Different types of fuzzy classifiers have been used. One type of classifier (FR1) uses graphs like histogram to generate an antecedent membership function and attributes are divided into several fuzzy sets. Second type uses a rule generation based on partition of overlapping areas (FR2). In addition to previously mentioned classifiers, another method uses a neuro-fuzzy computing framework neural-network learning paradigms are used to model newer fuzzy inference systems.

- **Hidden Markov Model (HMM):**

This technique is a stochastic version of the state-transition scheme. In this technique, states and transition probabilities are modeled as a Markov process with unknown parameters. A learning phase estimates these unknown parameters from

the input data. It is applied to detect anomalous traces of system calls in privileged processes. However, correct classification may not be possible if we use only system calls because in such cases many high level (connection features) are ignored. Further, HMMs are generative systems drawing joint probability distributions and fail to model long-range dependencies between the observations.

- **Support Vector Machines (SVM):**

The main disadvantage of neural networks is that they require large amount of data for training and it is often hard to select the best possible architecture for a neural network. SVMs are models that create the best decision boundaries to split the dataset. The boundaries are tend to split the data in a way that data is quite distant from the boundary and there is no other boundary possibly which holds the same conditions for each split of data. The boundary is more technically referred to as hyperplane. The nearest points to this boundary on each sides of data partitions are called as support vectors. Support vector machines find their applications in detecting intrusions in a system. Support vector machines are a better choice if we are dealing with data with large dimensions. They can be used for two class classifications as well as for multiple classes. SVMs can be a better choice for IDSs as they can enhance real time detection by map real-valued input feature vector to a higher dimensional feature space through nonlinear mapping.

Support vector machines (SVM) have also proven to be a better choice in detecting intrusions because of its training speed and scalability **[17]**.Several hybrid approaches for modeling IDS have been also explored. Decision Trees (DT) and Support Vector Machines (SVM) are combined as a hierarchical hybrid intelligent system model (DT- SVM).

- **Swarm Intelligence Algorithms:**

The swarm intelligence algorithm fully uses sensors that stochastically move around the classification habitat following pheromone concentrations. Having that aim in mind, a self-organized ANT colony based Intrusion Detection System (ANTIDS) is used to cluster the intrusion patterns **[18].**

- **Multivariate Adaptive Regression Splines:**

It is a nonlinear model that is used for regression. It uses various linear functions called base functions and predicts nonlinearities and variable dependencies. It is an innovative approach that automates the building of accurate predictive models for continuous and binary dependent variables .

- **Honey Pots:**

It is an unreal network system designed to trap crackers and intruders .The honeypot is used as lure in the form of a vulnerable system to trap hackers and keep them away from accessing the critical information in the system. In this technique alarming adversaries, initially detected by the IDS, will be rerouted to a honeypot network for a more close investigation. The connections will be handed over to the original destination if the investigations detect no abnormalities but will be terminated if investigations find an abnormal behaviour and an alarm will be raised. This action is hidden to the user. Such a scheme significantly decreases the alarm rate and provides a higher performance of IDS. This however is a very slow process and may not be useful in real time environments.

4. Types of Intrusion Detection System Based on Analyzing resources

Different IDSs have different approaches to finding intrusions in systems, the choice of which one to use depends on the overall risks to the organization and the resources available. IDS extensively examine the system so as to detect changes that it considers out of normal routine. These changes in the system resources can then be used to confirm the validity of alerts, investigate incidents, and correlate events between the IDS and other logging sources.

1. Network Based Intrusion Detection System (NIDS).

A network intrusion detection system works on network level and examines the packets that pass through a particular network. An NIDS exists as a software process on a dedicated hardware system.NIDSs are the most commonly used intrusion detection systems. An NIDS exists as a software process on a dedicated hardware system. A NIDS doesn't categorize the data and places the network interface card on the system into promiscuous mode, meaning that the card passes all traffic on the network to the NIDS software. It analyzes network traffic at all layers of the TCP/IP model and makes decisions about the purpose of the traffic [Intrusion Detection System]. It detects attacks by capturing and analyzing network packets. A Network-based intrusion detection system can monitor the network traffic affecting multiple hosts by listening to a network segment or a switch, thereby protecting those hosts.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

Various hosts acting as sensors are placed at different points in a network to ensure every packet is analyzed. These units monitor network traffic, performing local analysis of that traffic and report these attacks to a central management console. As the sensors are limited to running the IDS, they can be more easily secured against attack. To make NIDSs more efficient, many of these sensors are designed to run in "stealth" mode, in order to make it more difficult for an attacker to determine their presence and location.

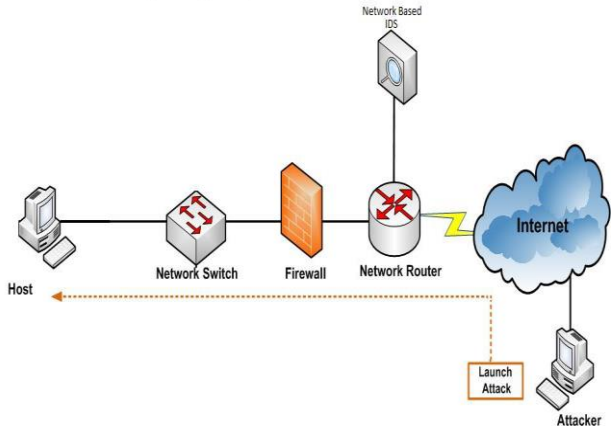

Figure 1.2 A Network Based Intrusion Detection System

Advantages:

- Network based IDSs can monitor large networks.
- The deployment of network-based IDSs has little impact upon an existing network.
- Network-based IDSs are usually passive devices that listen on a network wire without interfering with the normal operation of a network. Thus, it is usually easy to upgrade a network to include network-based IDSs with minimal effort.
- Network-based IDSs can be secured against attacks and even made invisible to many attackers.

Disadvantages:

- Network-based IDSs may not work efficiently in networks with high data flow, therefore, may fail to recognize an attack launched during periods of high traffic. This is usually a characteristic of software based NIDSs and, therefore, attempts are being made to solve this problem by implementing IDSs completely in hardware, which is much faster. The effectiveness of detection is drastically reduced as need to analyze packets quickly also forces vendors to both detect fewer attacks and also detect attacks with as little computing resource as possible.
- Modern switch- based networks don't rely much on the efficiency of Network based intrusion detection systems. Switches subdivide networks into many small segments (usually one fast Ethernet wire per host) and provide one to one connection between hosts serviced by the same switch. Universal monitoring ports aren't available on most modern switches and this limits the monitoring range of a network-based IDS sensor to a single host. We may not be able to mirror all traffic traversing traffic on a switch even if we get switches with such monitoring ports.
- Network-based IDSs cannot analyze encrypted information. VPNs are becoming vastly popular in many organizations and this has added more to the worry.
- Most network-based IDSs cannot tell whether or not an attack was successful; they can only discern that an attack was initiated. After a network-based IDS detects an attack, system administrators manually investigate each attacked host to determine whether the security policies of a system were compromised or not.
- Fragmentation of packets may cause some network-based IDSs to behave abnormally. These malformed packets cause the IDSs to become unstable and in due course, crash.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

2. Host Based Intrusion Detection Systems.

Host-based IDSs operate on information collected from the system log of individual hosts. The reliability and precision of HIDS is more as the host is being analyzed for activities, determining exactly which processes and users are involved in a particular attack on a particular operating system. A HIDS must be installed on each machine and requires configuration specific to that operating system and software [Intrusion Detection Systems]. Host-based IDSs has an advantage over Network based intrusion detection systems as they can estimate the outcome of an attempted attack, as the data files and system processes usually targeted by attacks are under direct control of the system.

More recently, a new form of HIDS has been created that examines calls to the operating system kernel . This type of HIDS is programmed with known attack signatures and will raise an alarm if a system call matches any of the signatures. Another HIDS for cloud computing has been proposed by **[19]**. Both types of HIDS are capable of checking files on the system for modification. This is done by performing a cryptographic checksum on the file using a hashing function such as MD5. This value is then stored and used as a comparison against periodic checksums of the file. If the checksums do not match, the file has been altered and the HIDS will report this information.

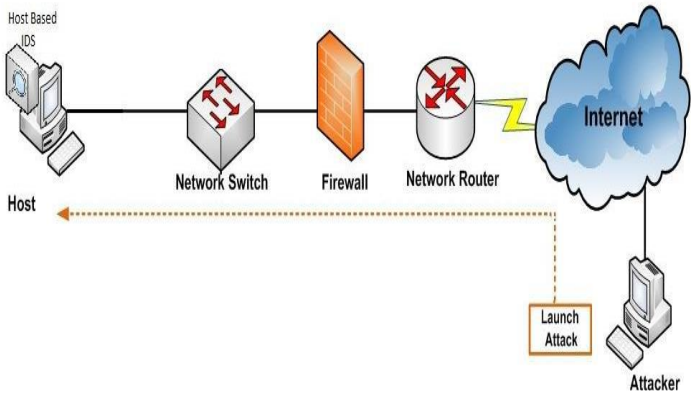

Figure 1.3 A Host Based IDS

Host-based IDSs normally utilize information sources of two types, operating system audit trails, and system logs. Operating system audit trails are usually generated at the innermost (kernel) level of the operating system, and are therefore, more detailed and better protected than system logs. However, system logs are much less obtuse and much smaller than audit trails, and are furthermore far easier to comprehend. Some host- based IDSs are designed to support a centralized IDS management and a reporting infrastructure that can allow a single management console to track many hosts. Others generate messages in formats that are compatible with network management systems.

Advantages:

- Host-based IDSs, with their ability to monitor events local to a host, can detect attacks that cannot be seen by network-based IDS.

- Host-based IDSs can often operate in an environment in which network traffic is encrypted, when the host-based information sources are generated before data is encrypted and/or after the data is decrypted at the destination host.

- Host-based IDSs are unaffected by switched networks or segmented networks.

- When Host-based IDSs operate on OS audit trails, they can help detect Trojan-Horse or other attacks that involve software integrity breaches. These appear as inconsistencies in process execution.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

Disadvantages:

- Host-based IDSs are harder to manage, as information must be configured and managed for every host monitored.

- Since at least the information sources (and sometimes part of the analysis engines) for host-based IDSs reside on the host targeted by attacks, the IDS may be attacked and disabled as part of the attack.

- Host-based IDSs are not well suited for detecting network scans or other such surveillance that targets an entire network, because the IDS only sees those network packets received by its host.

- Host-based IDSs can be disabled by certain denial-of-service attacks.

- When host-based IDSs use operating system audit trails as an information source, the amount of information can be immense, requiring additional local storage on the system.

- Host-based IDSs use the computing resources of the hosts they are monitoring, therefore inflicting a performance cost on the monitored systems.

3. Network Behaviour Analysis Intrusion Detection System.

Network Behavior Analysis Detection examines traffic on network segments to identify threats that generate unusual traffic flows, such as distributed denial of service attacks, certain forms of malware (e.g., worms, backdoors), and policy violations (e.g., a client system providing network services to other systems). It requires several sensors to create a good snapshot of a network and requires benchmarking and base-lining to determine the nominal amount of a segment's traffic. These systems are most often deployed to monitor flows on an organization's internal networks, and are also sometimes deployed where they can monitor flows between an organization's networks and external networks (e.g., the Internet, business partners' networks).

4. Wireless-based Intrusion Detection System.

Wireless-based Intrusion Detection System monitors wireless network traffic and analyzes its wireless networking protocols to identify suspicious activity involving the protocols themselves. It cannot identify suspicious activity in the application or higher- layer network protocols (e.g., TCP, UDP) that the wireless network traffic is transferring. However, it will analyze wireless-specific traffic, including scanning external users who try to connect to access points (AP), rogue APs, users outside the physical area of the company, and WLAN IDSs built into APs **[Intrusion Detection Systems]**. As networks increasingly support wireless technologies at various points of a topology, WLAN IDS will play larger roles in security. Many previous NIDS tools will include enhancements to support wireless traffic analysis.

5. Application-Based Intrusion Detection System

Application-based Intrusion Detection Systems are a special subset of host-based IDSs that analyze the events transpiring within a software application. The most common information sources used by application-based IDSs are the application's transaction log files.

The ability to interface with the application directly, with significant domain or application-specific knowledge included in the analysis engine, allows application-based IDSs to detect suspicious behavior due to authorized users exceeding their authorization. This is because such problems are more likely to appear in the interaction between the user, the data, and the application **.**

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

Advantages:

- Application-based IDSs monitors the interaction between user and application, which often allows them to trace unauthorized activity by authorized users.

- Application-based IDSs can often work in encrypted environments, since they interface with the application at transaction endpoints, where information is presented to users in unencrypted form.

Disadvantages:

- Application-based IDSs may be more vulnerable than host-based IDSs to attacks as the applications logs are not as well-protected as the operating system audit trails are used for host-based IDSs.

- As Application-based IDSs often monitor events at the user level of abstraction, they usually cannot detect Trojan-Horse or other such software tampering attacks. Therefore, it is advisable to use Application-based IDS in combination with Host- based and/or Network-based IDSs.

a. Protocol-based IDS

Typically protocol-based IDS are installed on a web server, and they are used for monitoring and analysis of the protocol in use of the computing system. If there is a deviation from intended behavior of protocol, then it can be detected as intrusion.

b. Graph-based IDS.

Graph-based IDS concerned with detecting intrusions that involve connections between many hosts or nodes. A graph consists of nodes representing the domains and edges representing the network traffic between them.

6. Hybrid IDS

The current trend in intrusion detection is to combine both host based and network based information to develop hybrid systems. In this type both kinds of IDS can be used simultaneously. This although is a very complex system and analyzing data on different levels may not be a viable solution especially in real time environments.

5. IDS-Challenges & Opportunities

In today's world, there seems to be drastic bend towards digitalization, which means that the information is being stored on computers. Thus all the communities either business or individuals depend on the computers for their information storage and processing, and if they want to share their critical/private information with anybody in any corner of the world, they rely on computer systems and networks. Thus, it became mandatory to store that information, and protect the networks, which carry that information from unauthorized users **[20]**. In order to have our network safe from these black hats, a new field has emerged in computer science and information security known as Intrusion detection and prevention system. Developed in late 80's, it began as a system that would generate reports related to attacks. Later it emerged as a tool for detecting different attacks, and simultaneously prevent them. The intrusion detection may be defined as a process of denying permission on some data to someone who does not have valid permission to access the same data. Thus we can say that intrusion is act of attempting access to some other's information/data without proper authorization or it is collection of actions on a system which violates security aspects (confidentiality, integrity, availability and authenticity) of a system's resources and on the other hand intrusion detection system is a process which detects these actions / violations of security on network data. The main purpose of the intrusion detection and prevention system is to review, control, analyze and produce reports from the system activities. Even though a lot of research has been done in the past in this particular field, there are still a number of issues and challenges it faces. The research communities are working very hard but it is dynamic and a vast field, and therefore, needs more research attention. The researchers have generally categorized the attackers into three different categories - insider, outsider and unknown. Also according to the report by

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

University of PEREAUS, the total numbers of insider attacks are 34%, while as for outsider attacks are 37% and rest 29% are unknown.

The challenge also lies in the fact that the developments in the field of hardware, software and other technologies does not occur in parallel. The newest advancements in these areas and the improvisation of intruders over their previous attacks, finding flaws in security models, etc. and thereby showing a novel behavior every time is in itself a challenge for these security systems. In some cases, the frequency of attacks has also proved to be fatal to the security of a computing system. Large amounts of events that occur in such cases, thus increasing the volume of audit trials, which in turn cause difficulties in identifying the abnormal behaviors in such large system logs.

In late 80's and early 90's many techniques were employed by IDS to hold back these intrusions ranging from statistical to empirical, but none of them has provided a final solution to this problem. In later years, the functionality of these techniques was also combined to reinforce the behavior of IDS's but we still lack accuracy and perfectness. Some techniques are not good with novel attacks, whereas some are difficult to program, and some produce large false alarms. In this chapter we will discuss the current issues and challenges that are faced by current IDSs.

5.1 General Architecture of an IDS.

A panoramic view of an IDS reveals that it is a security system that monitors a target system continuously and produces audit trials. These audit trials contain processed data generated from the information coming from the target system **.** The audit trials can then be inspected automatically by some tools either online or offline and/or used by some manual authority (an administrator) who analyzes these logs much closely. The general architecture of an IDS is shown in Figure 1.1. The location of an IDS is a significant issue and depends on various factors like the security level, budget and the environment. Generally, an IDS either is placed at the network entry/exit points or with hosts itself, or sometimes a combination of both. The job of an IDPs is to simply monitor the data, analyze it and accordingly prevent intrusions. The abnormal behavior detected by an IDPs system can be dealt automatically or by producing alarms to the manual station. An IDPs distinguishes between the normal and an abnormal behavior based on previous knowledge or policies already defined in it database.

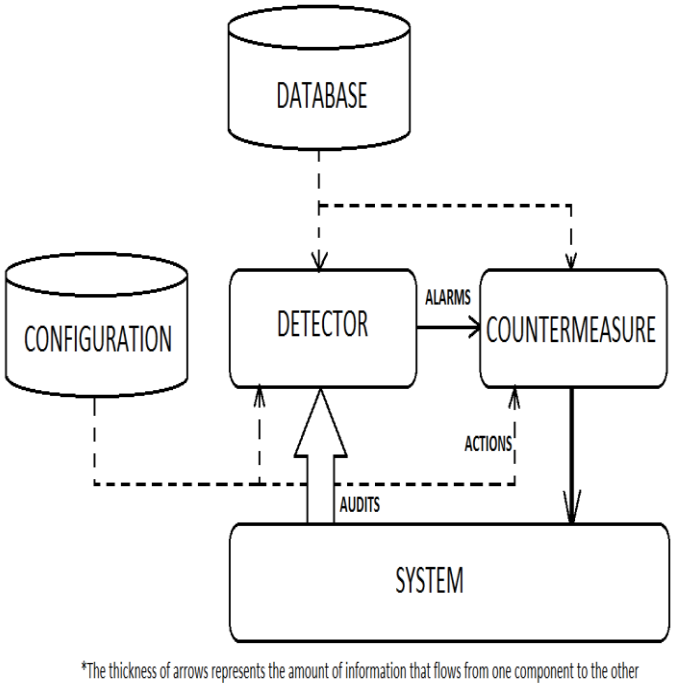

Figure 1.4 General Architecture of an IDS.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

5.2 Need for an Intrusion Detection System

As discussed in above section, intrusion can be defined as a process of accessing someone's personal data or information without proper privileges or rights. Since the data or information is widely available online through the Internet, or computer programs, this method of storing data increases the security risks in huge quantity. According to Symantec report, around 60,000 websites are available online, thus a person need to be no computer expert or a hacker. He can just download / run a hacking program (without no or slight modifications), configure some settings and he is done **[21]**. In order to secure data/information of an organization or an individual, firewalls were installed, but they do not serve the purpose of defending the data from attacks or intruders. The main aim of the firewall is to filter the traffic but they cannot block all the traffic. Also once the traffic passes through the firewall, there is no such mechanism available that will monitor the traffic for further processing. Also firewalls only monitors external traffic coming to it, but it doesn't detect the internal attacks. By using intrusion detection system, we can monitor or do the following things:

- Monitors network traffic.
- Monitor servers and network for continuous misuse actions or abuse policy.
- Attack/ breach, alerting, response and reporting.
- Provide counter measures

.

Thus, it became very much important for an organization to install both firewalls and intrusion detection systems to secure their assets / information from hackers / attackers.

5.3 Issues & Challenges in IDS

Today intrusion detection systems are still in infancy and need lot of research work to be done to make the intrusion detection even more successful. There are a huge number of issues and challenges in current intrusion detection system which needs the immediate and strong research attention. In this chapter, we have identified some important issues and challenges which need to be addressed by research communities. The issues and challenges are as:

- Deficiency or incomplete dataset.
- Detection algorithms.
- Integration of multiple formats of data.
- Platform dependencies.

Dataset

Dataset can be defined as a collection of all the data or information during recorde during the survey which needs to be analyzed. Since in intrusion detection systems, the datasets play an important role in accuracy of results, thus it became necessary to have datasets which are almost near to real time system. Nowadays, 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 emerged after this time period. Thus it became very much important that attack models should be tested in updated datasets. Therefore, this problem needs to be addressed in order to have most accurate and simplified results.

Detection Policy

This is the main part, which finds whether the packet/ information entering a system is an attack or useful information which might be needed by the user. The detection algorithm should be competent enough to analyses all the activities in minimum time that too it should do it in an efficient manner. 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. Hence we are in need of a good and fast algorithm which will detect the pattern thoroughly and fast to match the most of the attacks.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

Integration of Multiple formats

As we are well aware of the fact that the incoming frames or data may be in different formats. Therefore, there is a 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 **[22].**

Platform Dependence

In current technological world, we have different / number of intrusion detection systems available - some are free source while other are commercial. While implementing these intrusion detection systems all of them have some system requirements to implement that particular intrusion detection software. Therefore, it needs some platform for implementation. As we do have different platforms, we need an 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 make the changes as desired and re-deploy it again. Hence the design of the intrusion detection system must be like as :

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.

Testing and evaluation of IDS

As discussed above, data is growing enormously and IDSs has now become a standard for securing large network. Companies are investing huge amounts in IDS technologies, but there are no such scientific methods to test the 42 effectiveness of these IDS. Even though some quantitative measurable methods have been design to test the effectiveness, but they do not evaluate the effectiveness on the 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, they are not sufficient enough to figure out effectiveness of an IDS. Also there should be some common scale for evaluating or testing the effectiveness of IDS. The different issues are as **:**

- Collecting script and victim softwares.
- Different requirements for testing different types of IDS.
- Testing with different parameters.
- Poor design.
- Testing/evaluation of an IDS.

5.4 Theoretical Study of different IDS

Here in this section, we have made an attempt to carry out the comparative study of different intrusion detection technique. We have chosen some parameters on which the comparative study will be carried out. The parameters are in the table below:

Table-1 Parameters for comparative study of an IDS

Parameter	**Description**
Name	The name of the intrusion detection system
Type	The type of tool, or category in which this tool belongs, e.g., ―Web Application Scanning
Platform	The operating system(s) on which the tool runs. If the

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

	tool is an appliance, this field will contain a ―not applicable symbol (N/A) because the operating system is embedded in the tool.
License	The type of license under which the tool is distributed, e.g. Commercial, Freeware, GNU Public License
Based on	The technology on which IDS is based on i.e. rule based, pattern matching etc.
Suitability	On what kind of networks or systems it will be best implemented
Attacks detected	What kind of attacks is detected by the system?

The comparative studies based on the parameters discussed in table-1 are shown in below mentioned table-2 in details:

Name	**Type**	**Platform**	**License**	**Based on**	**Suitability**	**Attacks detected**
AIDE-Advanced Intrusion Detection Environment	HIDS	Linux 2.6, Solaris 10/Open Solaris, FreeBSD 2.2.8, 3.4, Unixware 7.0.1, BSDi 4.1 , OpenBSD 2.6, 3.0, AIX 4.2 , TRU64 4.0x, HP-UX 11i, Cygwin	Open source	Rule based	Suitable for checking the integrity of file and directory, mainly used for security purposes and can be used in small, medium a, large scale organizations, suitable in Linux and Unix based environments	File and Integrity checker
CSP Alert Plus	HIDS	Windows	Commercial	Rule based	Suitable for checking integrity of file and directory. Mainly useful for security purpose and can be used in large scale organizations	Intrusion, file and integrity checker.
Snort	NIDS	Linux	Open source	Rule based	Suitable for checking intrusions or	DOS & CGI attacks,

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

					attacks for large or small organizations	Intrusion attacks, port scans, Server message probes layer3 and related attacks.
Bro	NIDS	Linux	Open source	Pattern matching	Suitable for checking intrusions in the system for known attacks.	Signature inspection based.
AAFID	NIDS	Windows NT, Linux, FreeBSD, Open BSD	Open source	Statistical based	Suitable for checking intrusions or attacks for large or small organizations	DOS, File system attacks
DTK	HISD	Free BSD, Open BSD Linux, MAC OS	Open source	Statistical based	Works as a deception to attackers and is suitable for Linux and Unix based environments	Resource exhaustion, port scanning
ImSafe	NIDS	Free BSD, Open BSD Linux, MAC OS	Open Source	Statistical based	Suitable for buffer overflow attacks and react in real time, monitors system calls in Linux and Unix based systems. It is suitable for small scale organizations.	Buffer overflow stack.
Host-S Entry	HIDS	Linux, Free BSD	Open source	Statistical based	Suitable for detecting login anomaly detection, traces suspicious	Unknown user logins, suspicious user activities, Suspicious

					user activity, monitors interactive login sessions, and reports or reacts in real time in Linux based systems. It is suitable for environments where authorization and authentication is a main concern	user login domains.

Intrusion detection is used for securing the private data of an individual or a company. Companies are investing huge amounts of money for securing their valuable assets or information. But there are number of issues and challenges present in today's intrusion detection system. Here, we discussed the objectives behind using the intrusion detection system and some important issues and challenges faced by current intrusion detection systems, which needs to be addressed by research communities.

6. Proposed Methodology

Here, we are going to mitigate some issues first by proposing theoretical model and then implementing the same. Ensemble learning is a machine learning algorithm that combines number of machine learning models resulting in models that can predict the results more accurately and efficiently with reduced error than the individual models from which these are built. Ensemble Methods are also known as Committee Methods. As Polikar and Zhang points out the real world, terming it to be part of civilized society right from its beginning e.g. democracies and judicial systems. The fact that I am writing this text and taking ideas and understandings from many books and papers, and not only relying on the description of a single author who might have understood it all also describes the existence of ensemble learning in our day to day actions. Mostly, ensemble learning is used to improve the classification, prediction, and approximation performance of a model, or reduce the likelihood of an ill selection of a poor one. This has been the case in a number of machine learning competitions, where the winning solutions used ensemble methods. In the popular Netflix Competition, the winner used an ensemble method to implement a powerful collaborative filtering algorithm. Another example is KDD 2009 where the winner also used ensemble methods. Research in ensemble is continuously evolving to design ensembles consisting of competent yet complementary methods.

Numerous empirical and theoretical studies have revealed that ensemble models frequently attain higher accuracy than single models. The members of the ensemble might be predicting real-valued numbers, class labels, posterior probabilities, rankings, clustering, or any other quantity. Therefore, their decisions can be combined by many methods, including averaging, voting, and probabilistic methods. The majority of ensemble learning methods are generic, applicable across broad classes of model types and learning tasks.

The principle of combining predictions has been of interest to several fields over many years. Over 200 years ago, a controversial question had arisen, on how best to estimate the mean of a probability distribution given a small number of sample observations. The sample mean was not always optimal: under a simple condition, the sample median was a

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

better combined predictor of the population mean. The financial forecasting community has analyzed model combination for several decades, in the context of stock portfolios. The contribution of the Machine Learning (ML) community emerged in the 1990s-automatic construction (from data) of both the models and the method to combine them. While the majority of the Machine Learning literature on this topic is from 1990 onward, the principle has been explored briefly by several independent authors since the 1960s.

The study of ensemble methods, with model outputs considered for their abstract properties rather than the specifics of the algorithm which produced them, allows for a wide impact across many fields of study. If we can understand precisely why, when, and how particular ensemble methods can be applied successfully, we will have made progress toward a powerful new tool for Machine Learning: the ability to automatically exploit the strengths and weaknesses of different learning systems.

6.1 Types of ensemble Learning

Ensemble learning can be achieved using either of the two learning methods-trainable combiners and non-trainable classifiers. In a trainable classifiers, the success of a classifier with a particular data space is known in advance. This means if the input to a classifier is feed from these samples, the efficiency of an ensemble can be increased. In a non-trainable combination, the weight of the final classifier is known and the classifier which led to the efficiency of that 92 sample set remains unknown. The figure below depicts the different types of ensemble techniques based on these combination methods:

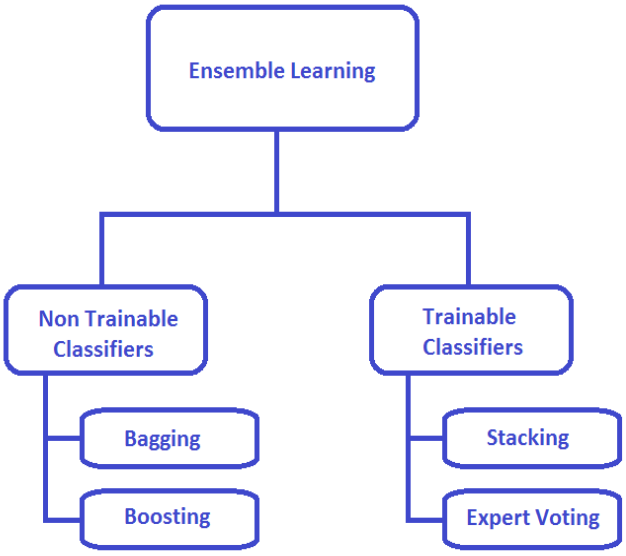

Figure 1.5 Types of Ensemble learning

Typically, an ensemble is constructed in two steps. First, a number of base learners are produced, which can be generated in a parallel style or in a sequential style where the generation of a base learner has influence on the generation of subsequent learners. Then, the base learners are combined to use, where among the most popular combination schemes are majority voting for classification and weighted averaging for regression.

Generally, to get a good ensemble, the base learners should be as more accurate as possible, and as more diverse as possible. There are many effective processes for estimating the accuracy of learners, such as cross-validation, hold-out test, etc. However, there is no rigorous definition on what is intuitively perceived as diversity. Although a number of diversity measures have been designed which disclosed that the usefulness of existing diversity measures in constructing ensembles is susceptible. Different base learners can be introduced from different channels, such as sub-sampling the training examples, manipulating the attributes, manipulating the outputs, injecting randomness into learning algorithms,

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

or even using multiple mechanisms simultaneously. The employment of different base learner generation processes and/or different combination schemes leads to different ensemble methods.

There are many effective ensemble methods. The following will briefly introduce three representative methods - Boosting **[23],** Bagging **[24]** and Stacking **[25],** Here, binary classification is considered for simplicity. That is, let X and Y denote the instance space and the set of class labels, respectively, assuming Y = {Normal, Anomaly}. A training data set D = {(x1, y1), (x2, y2),…,(xm, ym)} is given, where xi ε X and yi ε Y (i = 1,…,m).

a. Bagging

Bagging (short for bootstrap aggregating) is a simple and an effective technique which combines the classification model of various methods training each on an individual data set drawn from sampling the original data set. The final result of training is then combined using the average voting scheme. Bagging as introduced by Breiman in 1994 does not produce the most optimal of the classifiers, however, the process can be optimized by introducing the concept of order-correct classifier which can produce better classification **.**

A bootstrap sample is obtained by subsampling the training data set with replacement, where the size of a sample is as the same as that of the training data set. Thus, for a bootstrap sample, some training examples may appear but some may not, where the probability that an example appears at least once is about 0.632. After obtaining the base learners, Bagging combines them by majority voting and the most-voted class is predicted. The pseudo-code of Bagging is shown below. It is worth mentioning that a variant of Bagging, Random Forests, has been deemed as one of the most powerful ensemble methods up to date.

b. Boosting

In this technique the training of individual classifiers differs from its earliest counterpart by the fact that the training subset of next classifiers usually contains the instances that were misclassified by the earlier classifier. It means as we go on building an ensemble incrementally, the new subset of training data will contains misrepresented instances classified by previous classifier. Usually, the original dataset is divided into three subsets.

The first random subset is fed to a classifier thereby building a training model. The new classifier contains a random subset of new training data in addition to the instances that were misclassified by first model. The third subset is constructed by combining the misclassified instances from both the previous classifiers. Finally, the results are also classified by the average voting scheme. AdaBoost is the most popular example of this technique.

First, it assigns equal weights to all the training examples. Then, designate the distribution of the weights at the t-th learning round as Dt. From the training data set and Dt the algorithm generates a base learner ht : X → Y by calling the base learning algorithm. Then, it uses the training examples to test ht, and the weights of the incorrectly classified examples will be increased. Thus, an updated weight distribution Dt+1 is obtained. From the training data set and Dt+1 AdaBoost generates another base learner by calling the base learning algorithm again. Such a process is repeated for T times, each of which is called a round, and the final learner is derived by weighted majority voting of the T base learners, where the weights of the learners are determined during the training process. In practice, the base learning algorithm may be a learning algorithm which can use weighted training examples directly; otherwise the weights can be exploited by sampling the training examples according to the weight distribution Dt.

C. Stacking

Also known as stacking generalization, this concept was introduced by David Wolpert in 1992 and uses the concept of layered ensemble learning. At the top level stands a meta learner which is supposed to achieve more predictive accuracy due to the learning from lower layer classifiers. The training data is initially divided in to ‘n’ subsets, each of them are fed to individual classifiers. Each classifier gets a different overlapped subset. The result of these classifiers is then input to a classifier model which sits in top of this layer. Additionally, a different subset of random training data can be input to this meta learner.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

By and large, there is no ensemble method which outclasses other ensemble methods unswervingly. Empirical studies on popular ensemble methods can be found in many papers such as **[26].** Previously, it was thought that using more base learners will lead to a better performance, yet Zhou et al. **[27]** proved the "many could be better than all" theorem which indicates that this may not be the fact. It was shown that after generating a set of base learners, selecting some base learners instead of using all of them to compose an ensemble is a better choice. Such ensembles are called selective ensembles.

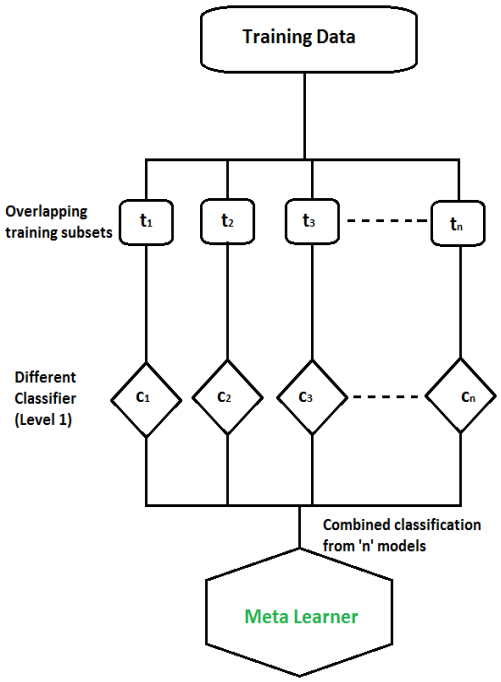

Figure 1.6 Stacked Generalization

It is worth mentioning that in addition to classification and regression, ensemble methods have also been designed for clustering and other kinds of machine learning tasks.

6.2 Error in Ensemble Learning (Variance Vs Bias)

Mathematically, the error emerging from any model can be broken down into three components. Following are these components:

$$Err(x) = \left(E[\hat{f}\,(x)] - f(x)\right)^2 + E\left[\hat{f}\,(x) - E[\hat{f}\,(x)]\right]^2 + \sigma_e^2$$

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

Bias error is useful to quantify how much on an average are the predicted values different from the actual value. A high bias error means we have an under-performing model which keeps on missing important trends.

Variance on the other side quantifies how are the prediction made on same observation different from each other. A high variance model will over-fit on your training population and perform badly on any observation beyond training. Following diagram will give you more clarity (Assume that red spot is the real value and blue dots are predictions):

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

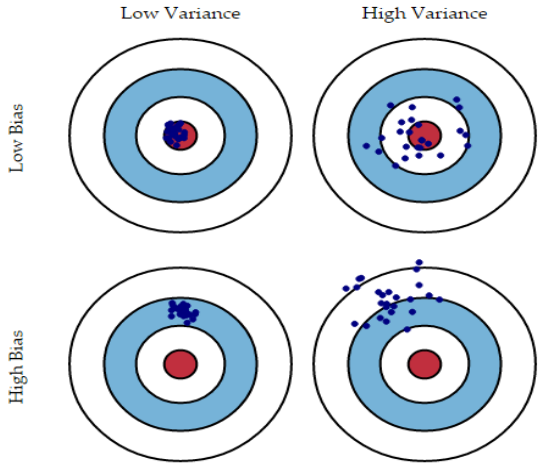

Figure 1.7 Graphical representation of variance and bias

Normally, as we raise the complexity of our model, we will see a reduction in error due to lower bias in the model. However, this only happens till a particular point. As we continue to make our model more complex, we end up over-fitting our model and hence our model will start suffering from high variance.

A champion model should maintain a balance between these two types of errors. This is known as the trade-off management of bias-variance errors. Ensemble learning is one way to execute this trade off analysis.

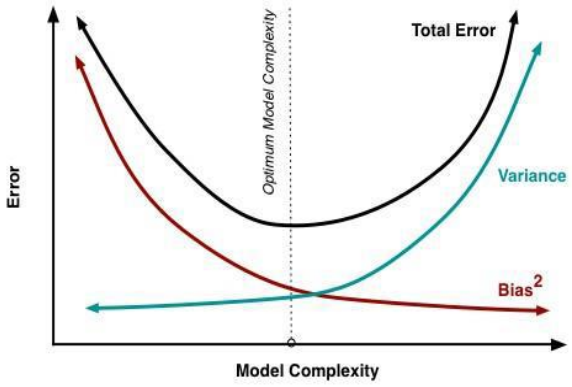

Figure 1.8 Trade-off management of bias-variance errors

7. Proposed Ensemble Model

The dataset which was reduced to a set of most prominent features based on experiment on three parameters were input to our proposed empirical/theoretical model. The results of individual classifiers was then combined through majority voting scheme to prepare a final classifier that is ready to be tested on the real data. In our model we have used three base classifiers. Our descriptive and comparative study helped us to short list the three algorithms. We have also taken a technique from the theory of induction through decision trees which is considered to be an ensemble in itself. Hence the overall accuracy of random trees is already better. This has helped to improve the overall accuracy of the proposed system. The two base learners are not as accurate as standalone algorithms. In our effort to make better ensemble and in turn a better IDS, we tried many variations of combinations. The final proposed model turned out to be more accurate and the time taken was also reduced by a large extent. Our extensive efforts were also bound by limitations of time and the resources e.g. the hardware and the software that was available to us.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1844 - 1865
https://publishoa.com
ISSN: 1309-3452

The diagram below gives a graphical view of how the ensemble was tested on the original dataset and then the optimized dataset.

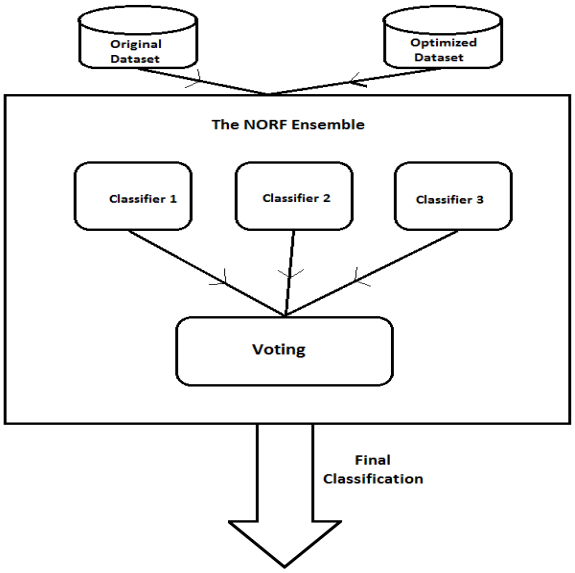

Figure 1.9 The Proposed ensemble model

8. Result.

Confusion Matrix

	Positive	**Negative**
Positive	66635	708
Negative	5976	52654

Classifier	**True Positive Rate**	**False Positive Rate**	**Mean Absolute Error**	**Root mean squared**	**Time Taken**
Proposed ensemble Model	0.947	0.059	0.0801	0.1973	149.65
Correctly Classified Instances			119289	94.6941 %	
Incorrectly Classified Instances			6684	5.3059 %	

The results in the above table provide the optimized results that were generated from the proposed model. The model has a near 95% accuracy which is much higher than what the sub-classifiers in this model could have performed individually. The time taken to generate the model is also very less which means such a model would quickly train itself, to prepare itself for the novel attacks. In the real world, acting against a threat quickly takes more precedence that trying to be more

accurate every time. Even the top most security agencies in this world are not perfect with all the human brains working with artificial intelligence. We have although tried to achieve the near accurate results in a lesser amount of time.

The false alarm rates have also been reduced so as to make alarms more reliable. Since the model is based on techniques that use artificial intelligence, this factor needed to be drastically reduced. The greater the errors in training the model leading to false alarms, the higher the chances of some novel attacks to break into a system without being detected. The false alarm rate has been drastically reduced to 5% in most of the cases. The individual attack types were also checked on the model and some of them were even reduced to less than 3%.

9. Conclusion

With machines taking over the most complex jobs in the world, it seems that in future, the humans have to deal with a parallel computer-based ecosystem. The pervasive computing and the fast Internet connections and their applications altogether have already pushed humans towards relying more on automated systems. Since network is the core to all this, we see every other organization or an individual acquainting himself with the Internet.

Security is always a major concern when people put their money or assets online. Since a network is an open road for packets (in essence data), unwanted users always try to make illegal use of these networks. Many important measures like authentication, authorization and use of firewalls were used in early stages to prevent such illegal access into systems. Intrusions Detection Systems have been developed overtime as a major security tool. Since then many intrusions have been detected and gradually acted upon but they continue to evolve and develop still. Advancements in the field of artificial intelligence has given a different face to IDSs.

Security researchers face many problems in developing tools which are flawless in their behavior towards intrusions. In the earlier chapters, we have learned about the limitations in making a secure system. The non-availability of data to be researched upon is the first and the foremost issue. The security agencies need to provide newer data so that we can learn from every day. There are some attacks which we are still unknown to us because of the security policies of certain organizations. Lack of data, in addition to other factors is a major reason for building a stronger and more secure system. Besides, loopholes in both software and hardware platforms continue to provide an opportunity for the intruders to barge into our systems. Intruders fully acquaint themselves with the hardware and software bugs and use these weak points to break into these systems.

We also have to understand the fact that artificial intelligence though has been in the field for a long time now but the ideas and techniques need to be refined more. Studies continue to be done to improve the algorithms and make them more responsive in real time and more accurate. We have already mentioned in the previous chapters about the false alarms and other related issues that make the existing techniques used by IDSs unreliable.

Our research will continue with the use of latest algorithms or newer versions of older techniques to improve our model. The prime issue continues to be saving the world from attacks and threats by outsiders and those people who don't belong in a particular system. Developing IDSs over distributed environments will also be helpful as the detection engine will use the power of many machines simultaneously. These systems have already been started but using artificial algorithms on a distributed environments remains an active area of research.

References

1. Russel, D., Gangemi, G.T. "Computer security basics," CA: O=Reilly & Associates Inc. 448p. 1992.
2. Caelli, William, Dennis Longley, and Michael Shain. Information security handbook. Stockton Press, 1991.
3. Anderson, James P. Computer security threat monitoring and surveillance. Vol.17. Technical report, James P. Anderson Company, Fort Washington, Pennsylvania, 1980.
4. Mukkamala, Srinivas, Andrew H. Sung, and Ajith Abraham. "Intrusion detection using ensemble of soft computing paradigms." Intelligent systems design and applications. Springer Berlin Heidelberg, 2003. 239-248.

5. Chebrolu, Srilatha, Ajith Abraham, and Johnson P. Thomas. "Feature deduction and ensemble design of intrusion detection systems." Computers & security 24.4 (2005): 295-307.
6. Peddabachigari, Sandhya, et al. "Modeling intrusion detection system using hybrid intelligent systems." Journal of network and computer applications 30.1 (2007): 114-132.
7. Zainal, Anazida, Mohd Aizaini Maarof, and Siti Mariyam Shamsuddin. "Ensemble classifiers for network intrusion detection system." Journal of Information Assurance and Security 4.3 (2009): 217-225.
8. Giacinto, Giorgio, et al. "Intrusion detection in computer networks by a modular ensemble of one-class classifiers." Information Fusion 9.1 (2008): 69-82.
9. Lee, Wenke, Salvatore J. Stolfo, and Kui W. Mok. "A data mining framework for building intrusion detection models." Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on. IEEE, 1999.
10. Abraham, Ajith, Crina Grosan, and Yuehui Chen. "Evolution of intrusion detection systems." School of Computer Science and Engineering, Chung-Ang University, Korea (2005): 2-3.
11. de Lima, Igor Vinicius Mussoi, Joelson Alencar Degaspari, and Joao Bosco Mangueira Sobral. "Intrusion detection through artificial neural networks." Network Operations and Management Symposium, 2008. NOMS 2008. IEEE. IEEE, 2008.
12. Owais, Suhail, et al. "Survey: using genetic algorithm approach in intrusion detection systems techniques." Computer Information Systems and Industrial Management Applications, 2008. CISIM'08. 7th. IEEE, 2008.
13. Visumathi, J., and K. L. Shunmuganathan. "A computational intelligence for evaluation of intrusion detection system." Indian Journal of Science and Technology 4.1 (2011): 40-45.
14. Amor, Nahla Ben, Salem Benferhat, and Zied Elouedi. "Naive bayes vs decision trees in intrusion detection systems." Proceedings of the 2004 ACM symposium on Applied computing. ACM, 2004.
15. Jemili, Farah, Montaceur Zaghdoud, and Mohamed Ben Ahmed. "A framework for an adaptive intrusion detection system using bayesian network." Intelligence and Security Informatics, 2007 IEEE. IEEE, 2007.
16. A. El-Semary, J. Edmonds, J. Gonzalez and M. Papa, "A Framework for Hybrid Fuzzy Logic Intrusion Detection Systems," 14th IEEE International Conference on Fuzzy Systems, May 2005, pp. 325-330. doi:10.1109/FUZZY. 2005.1452414.
17. Mukkamala, Srinivas, Andrew H. Sung, and Ajith Abraham. "Intrusion detection using ensemble of soft computing paradigms." Intelligent systems design and applications. Springer Berlin Heidelberg, 2003. 239-248.
18. Ramos, Vitorino, and Ajith Abraham. "Antids: Self organized ant-based c lustering model for intrusion det ection system." Soft Computing as Transdisciplinary Science and Technology (2005): 977-986.
19. Deshpande, Prachi, et al. "HIDS: A host based intrusion detection system for cloud computing environment." International Journal of System Assurance Engineering and Management 9.3 (2018): 567-576.
20. Abadeh, M. Saniee, Jafar Habibi, and Caro Lucas. "Intrusion detection using a fuzzy genetics-based learning algorithm." Journal of Network and Computer Applications 30.1 (2007): 414-428.
21. Garcia-Teodoro, Pedro, et al. "Anomaly-based network intrusion detection: Techniques, systems and challenges." computers & security 28.1 (2009): 18-28.
22. Kandeeban, S. Selvakani, and Rengan S. Rajesh. "Integrated Intrusion Detection System Using Soft Computing." IJ Network Security 10.2 (2010): 87-92.
23. Schapire, Robert E. "The strength of weak learnability." Machine learning 5.2 (1990): 197-227.
24. Breiman, Leo. "Bagging predictors." Machine learning 24.2 (1996): 123-140.
25. Wolpert, David H. "Stacked generalization." Neural networks 5.2 (1992): 241- 259.
26. Bauer, Eric, and Ron Kohavi. "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants." Machine learning 36.1 (1999): 105-139.
27. Zhou, Zhi-Hua, Jianxin Wu, and Wei Tang. "Ensembling neural networks: many could be better than all." Artificial intelligence 137.1-2 (2002): 239-263.