A Project Report

On

ANOMALY DETECTION IN NETWORK

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE

By:

SHRISTI (100180149)

ANMOL SINGH (100180110)

SHIKSHA UPADHYAY (100180147)

UNDER THE GUIDANCE OF

Dr. AMIT SHARMA

(Asst. Professor, Computer Science & Engineering)

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

SIR CHHOTU RAM INSTITUTE OF ENGINEERING & TECHNOLOGY

CHAUDHARY CHARAN SINGH UNIVERSITY, MEERUT

(2021-22)

SIR CHHOTU RAM INSTITUTE OF ENGINEERING & TECHNOLOGY
CHAUDHARY CHARAN SINGH UNIVERSITY MEERUT
Approved by A.I.C.T.E., New Delhi

Certificate

This is to certify that the project entitled "Anomaly Detection In Network " is submitted in partial fulfillment of the requirement of the degree of BACHELOR OF TECHNOLOGY (Computer Science) of **Sir Chhotu Ram Institute of Engineering and Technology, Chaudhary Charan Singh University Campus, Meerut(U.P.)**, under the supervision.

Dr. Amit Sharma
(Project Supervisor)

Ms. Ritu Sharma
(Project Co-ordinator)

SIR CHHOTU RAM INSTITUTE OF ENGINEERING & TECHNOLOGY
CHAUDHARY CHARAN SINGH UNIVERSITY MEERUT
Approved by A.I.C.T.E., New Delhi

Declaration

The Project report entitled "ANOMALY DETECTION IN NETWORK" is submitted by SHRISTI (100180149), ANMOL SINGH (100180110) and SHIKSHA UPADHYAY (100180147). It warrants its Acceptance as a prerequisite for the degree of Bachelor of Technology (Computer Science) of **Sir Chhotu Ram Institute of Engineering and Technology, Chaudhary Charan Singh University Campus, Meerut(U.P.)**, under the supervision.

Ms. Ritu Sharma
(Internal Examiner)

Mr. Narender Kumar
(External Examiner)

Er. Milind Singh
(Co-ordinator)
Deptt. Of Computer Science Engineering

ACKNOWLEDGEMENT

We extend our thanks to Prof. Neeraj Singhal Director and Er. Milind Singh Co-ordinator for giving us the motivation and guidance to complete this project.

We are deeply indebted to our project guide Dr. Amit Sharma, for the initial idea of the project and for all the guidance and encouragement he gave in the subsequent months. His supervision and co-operation at every stage helped us in successfully completing the project. Whatever intellectual effort may be reflected from this project is the direct result of the informative and stimulating discussions and suggestions he has given during the course pf the project.

Lastly, we would like to thank the entire staff of the computer science department and our college for their support and co-operation in the completion of this project.

(Student Signature)

Student Name	Roll No.	Email Id
Shristi	**100180149**	**shristi.s1010@gmail.com**
Anmol Singh	**100180110**	**anmol.write@gmail.com**
Shiksha Upadhyay	**100180147**	**shikshau8@gmail.com**

Table of Contents

1. ABSTRACT

There has been rapid advancement in technology and networks in recent decades, and also the spread of internet globally. Because the number of pirated goods has risen and many modern systems have been hacked, developing network security technologies to identify new attacks has become a must. One way to secure a network against cyber crimes is definitely to have an 'Intrusion Detection System'.

A Intrusion Detection System employs ML and also DL techniques to find anomaly in the network. Focus here is to use a machine learning algorithm to find any suspicious thing using an Intrusion Detection System with good performance. Hence, in today's network communications, network-intrusion is the most alerting concern. Network-attacks are increasing rapidly and becoming very active with each passing day, which possess threats to network-services. Many researches have been conducted to solve this issue and to find a good solution to stop network intrusion and have safety. Machine learning algorithm is an effective analysis tool and is utilized in detecting anomalous network traffic flow events.

We proposed IDS framework based on unsupervised and supervised machine learning methods on cloud platform. The unsupervised machine learning method is used to make clusters while the supervised machine learning is used for classifying attack types. Proposed IDS give flexibility to the user on cloud for IDS deployment. It can handle problem of single point of failure. We have conducted three experiments. In the first experiment, same labeled clusters of different cloud users are merged, while in the second experiment, all clusters of same cloud user are combined before applying supervised learning method. In the third experiment, we have simulated attack from on vm to another vm of cloud and executed proposed framework on it. The experimental result shows that the proposed model improves the ability of intrusion detection.

2. INTRODUCTION

With the increasing graph of technological services and the rapid growth of internet globally, cases of cyber crimes and attacks have alsosoared. According to a report by Internet Security Threat Report (ISTR) in the year 2015 there were 420-million growing type of malwares were discovered, out of them, 362 are crypto ransom wares. If there is one crystal and true conclusion in 2022, which is: none of the companies and institutes are safe-to-use from cyber-attacks. With time cyber-attack has become progressed and abstract unlike earlier. Thus, security-techniques should also be updated and advanced continuously. As per Internet-Security-Threat-Report i.e., ISTR, in 2015 around 420-million growing types of malwares were found, including 363 Crypto-ransom- ware. If there is a final assumption in 2019, is that none of the companies are harmless from cyber attacks.

The network-intrusion-detection systems play a critical part in terms of privacy of network, by detecting invasion and delivers to the respective authorities. As per detection technique, there exists 2 types of IDS: Anomaly-detection & Misuse-detection:

Anomaly-Detection: It is used to make a usual action databases & other abnormalities from usual activity is observed where interference is conceivable in the network.

Misuse-Detection: It is used to define attacks-activity in the databases & in case, there's present similar types of opportunities in the networks and hence acknowledged as attacks.

When total number of cyber crime kinds grow, the anomaly-detection-system surpasses misuse-detection-system in terms of constructing a network intrusion detection system. A system that detects anomalies is better suited to detecting unknown attacks. In accordance with misuse- detection-system and anomaly-detection-system, many artificial intelligence (AI) methods have been proposed.

However, most rule-based IDSs have some drawbacks. They are unable to find growing attacks which are using different signature because these signatures are not inclusive in the knowledge-base. New grasping skills have been proposed to overcome the limitations. In addition, network security is a big safety concern to neutralize unnecessary activity. It's very important for security of data(s) & network-privacy as well as for ignoring possible danger situation. As per reports by Microsoft-Security-Intelligence from February to July 2010, virus trends have been on a rise at fast pace around the globe.

Cyber attacks happen all the time, affecting the internet in terms of security and privacy. Thus the security-system must be strong, reliable, and perfectly-designed. There are primarily two types of network intrusion detection. The first is signature based structure, and second being an anomaly based structure. A signature based detection-system looks through system-traffic for set of bytes or the packet sequencethat have previously been flagged as suspicious.

The disadvantage of this scheme is that if the attacker knows about what network behaviour to identify than signatures are easier to develop. Signature-based type detection has some drawbacks as well. Each attack requires the creation of a signature, and they can only detect those attacks. They can't detect any other new attacks because their signatures aren't recognized by the detection scheme. The concept of an anomaly based detection scheme depends on analyzing the network behaviour characteristic. So, this type of detection is capable of finding typical behaviour by analyzing high network traffic, or to a specific hosts, and networks load-imbalance. One disadvantage of this type of scheme is that it is not detected as an anomaly if the malicious behaviour falls within normal network behaviour.

A major advantage of using an anomaly-based detection in place of signature-based detection is that it can detect a new attack for which no signature exists if it behaves differently than normal traffic behaviour patterns. For ensuring data confidentiality, network security, classified data security, and

preventing unauthorized access to data detection of intrusion is a critical task. For network intrusion detection, a variety of methods have been proposed. A major component of network security is anomaly network intrusion detection. The anomaly's behaviour can sometimes appear to be the same as normal data usage.

Objectives

- The below objectives are anticipated to be achieved from the conclusion of the study:
- To look at the past work completed in the scope by making broadfield research.
- Choosing the fitting data set by making thorough examination on theoptions to the data set.
- Selecting reasonable calculations by focusing a large amount of research on AI calculations.
- Settling on right calculations by performing comprehensive examination on AI techniques.
- Choosing an appropriate software platform.
- Picking the reasonable equipment/gear stage.
- Settling on the right assessment rules.
- Picking the benchmark studies to be analyzed during the assessmentstage.

2.1 Brief history of IDS

The idea of detecting the intrusions or system misuses by looking at some kind malicious patterns in the network or user activity was initially conceived by James Anderson in his report titled “Computer Security Threat Monitoring and Surveillance” [2] to US Air Force in the year 1980. In the year 1984, the first prototype of Intrusion Detection System which monitors the user activities, named “Intrusion Detection Expert System” (IDES) was developed. In the year 1988,

"Haystack" became the first IDS to use patterns and statistical analysis for detecting malicious activities, but it lacked the capabilities of real time analysis.

Meanwhile, there were other significant advances occurring at University of California Davis' Lawrence Livermore Laboratories. In the year 1989, they built a IDS called "Network System Monitor" (NSM) for analyzing the network traffic. This project was subsequently developed into IDS named "Distributed Intrusion Detection System" (DIDS). "Stalker" based on DIDS became the first commercially available IDS and influenced the growth and trends of future IDS. In the Mid 90's, SAIC developed "Computer Misuse Detection System" (CMDS), a host based IDS. US Air Force's Cryptographic support centre developed "Automated Security Incident Measurement" (ASIM), which addressed the issues like scalability and portability. The intrusion detection market began to gain in popularity and truly generate revenues around 1997. In that year, the security market leader, ISS, developed a network intrusion detection system called "Real Secure". A year later, Cisco recognized the importance of network intrusion detection and purchased the Wheel Group, attaining a security solution they could provide to their customers. Similarly, the first visible host-based intrusion detection company, Centrax Corporation, emerged as a result of a merger of the development staff from Haystack Labs and the departure of the CMDS team from SAIC. From there, the commercial IDS world expanded its market-base and a roller coaster ride of start-up companies, mergers, and acquisitions ensued. Martin Roesch, in the year 1998 launched a light weight open source Network IDS named "SNORT" [3], which has since then gained much popularity. In year 1999 Okena Systems worked out the first Intrusion Prevention System (IPS) under the name "Storm Watch". IPS are the systems which not only detect the intrusions but also are able to react on alarming situation. These systems can co-operate with firewall without any intermediary applications.

2.2 Type of IDS

Depending upon the level of analysis IDS is classified into two major types:

Network based IDS (NIDS):

Monitors and analyzes the individual packets passing around a network for detecting attacks or malicious activities happening in a network that are designed to be overlooked by a firewall's simplistic filtering rules.

Host based IDS (HIDS):

Examines the activity on individual computer or host on which the IDS is installed. The activities include login attempts, process schedules, system files integrity checking system call tracing etc. Sometimes two kinds of IDS are combined together to form a Hybrid IDS.

Generally, IDS has two components –

Central Administration (Management) Module:

Provides centralized facility for managing and monitoring of all the installations of Intrusion Detection System and hence centralized way of analyzing and detecting the intrusions. It has the complete view of the various activities and events occurring in different segments of the organizational network. Moreover the policy settings, actions to be triggered, patches/signature updation, fine tuning of sensors can be achieved with this module.

IDS Sensors (Agents):

Analyses the network traffic and identifies attacks and security breaches, which take place by exploiting the technology of network implementation, reports the alerts to the Management module and performs the preset actions. IDS Agents are more autonomous in their functions as compared to the Sensors.

2.3 Detection technique

Various techniques are in place for intrusion detection which can be broadly classified as follows

Signature/pattern-based Detection:

In this technique, the sensors which are placed in different LAN segments filter and analyse network packets in real time and compares them against a database of known attack signatures. Attack signatures are known methods that intruders have employed in the past to penetrate a network. If the packet contents match an attack signature, the IDS can take appropriate countermeasure steps as enabled by the network security administrator. These countermeasures can take the form of a wide range of responses. They can include notifications through simple network management protocol (SNMP) traps or issuance of alerts to an administrator's email or phone, shutting down the connection or shutting down the system under threat etc. An advantage of misuse detection IDS is that it is not only useful to detect intrusions, but it will also detect intrusion attempts; a partial signature may indicate an intrusion attempt. Furthermore, the misuse detection IDS could detect port scans and other events that possibly precede an intrusion.

Unauthorised Access Detection:

In unauthorised access detection, the IDS detects attempts of any access violations. It maintains an access control list (ACL) where access control policies for different users based on IP addresses are stored. User requests are verified against the ACL to check any violations.

Behavioural Anomaly (Heuristic based) Detection:

In behavioural anomaly detection method, the IDS is trained to learn the normal behavioural pattern of traffic flow in the network over an appropriate period of time. Then it sets a baseline or normal state of the network's traffic, protocols used and typical packet sizes and other relevant parameters of network traffic. The anomaly detector monitors different network segments to compare their state to the

normal baselines and look for significant deviations.

2.4 IDS response against attack

Whenever IDS detects any intrusions or attacks, it reacts as per the preconfigured settings. The responses can range from mere alert notifications to blocking of the attacks based on the severity. The appropriate reactions on the threats are a key issue for safety and efficacy. Generally the responses can be of three types.

Active response:

IDS by itself cannot block attacks, however can take such actions which can lead to stopping of attacks. Such actions can be for example, sending TCP reset packets to the machine(s) which is being the target of attack, reconfiguring router/firewall as to block the malicious connection. In extreme cases, IDS can even block all the network traffic to avoid potential damage to the firm.

Passive response:

Passive solutions deliver information to IDS administrator on the current situation and leave the decision to take appropriate steps to his discretion. Many commercial systems rely on this kind of reactions. Examples for this kind of actions can be simple alarm messages and notifications. Notifications can be sent on email, cellular phone or via SNMP messages.

Mixed response:

Mixed responses combine both active as well as the passive responses Appropriately as per the needs of situation.

3.TECHNOLOGY USED

3.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

.

3.2.1 Libraries

3.2.2 Numpy Library

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

At the core of the NumPy package, is the *ndarray* object. This encapsulates *n*-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences:

- NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of an *ndarray* will create a new array and delete the original.
- The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sized elements.
- NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.
- A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays. In other words, in order to efficiently use much (perhaps

even most) of today's scientific/mathematical Python-based software, just knowing how to use Python's built-in sequence types is insufficient - one also needs to know how to use NumPy arrays.

3.2.3 OpenCV Library

It is abbreviated as cv2. When the library is retrieved, it is referred to as cv2. It is a free and open source library. It can recognise faces and objects in pictures and movies. It can even identify the object's handwriting. It does not perform well when utilized alone, but when combined with other Python libraries such as Numpy, it creates a highly efficient library for numerical calculations. Because it operates in the background, the Numpy library is required for OpenCV.

In OpenCV, the CV is an abbreviation form of a computer vision, which is defined as a field of study that helps computers to understand the content of the digital images such as photographs and videos.

The purpose of computer vision is to understand the content of the images. It extracts the description from the pictures, which may be an object, a text description, and three-dimension model, and so on. For example, cars can be facilitated with computer vision, which will be able to identify and different objects around the road, such as traffic lights, pedestrians, traffic signs, and so on, and acts accordingly.

Computer vision allows the computer to perform the same kind of tasks as humans with the same efficiency. There are a two main task which are defined below:

- **Object Classification -** In the object classification, we train a model on a dataset of particular objects, and the model classifies new objects as belonging to one or more of your training categories.
- **Object Identification -** In the object identification, our model will identify a particular instance of an object - for example, parsing two faces in an image and tagging one as Virat Kohli and other one as Rohit Sharma.

3.2.4 Time Library

Using Time library, we can handle various operations regarding time including its conversions which have wide use in many applications of day to day life. The beginning of the time measuring started on January 1 1970, 12:00 AM. This term is also called epoch. There are five operations of time:

1. time()
2. gmtime ()
3. asctime ()
4. ctime ()
5. sleep()

3.3.3 Jupyter Notebook

It is an open source web tool which enables programmers to create and share files including visualizations, equations, code, and text. The Project Jupyter team is in charge of maintaining this website constantly updated.

Jupyter Notebook is a fork of the IPython project, which previously had its own IPython Notebook project. Julia, Python, and R are the three major programming languages that Jupyter supports. The IPython kernel that comes with Jupyter allows us to write Python code. However, we now have access to more than 100 additional kernels.

Data cleansing and transformation, computational simulation, mathematical modelling, visualization of data, machine learning, and a variety of other tasks are all possible with it.

3.4 Machine Learning

Machine learning is the process in which computer algorithms are used based on the idea that the system can learn, adapt without following implicit instructions. In simpler words, it is a use of algorithms and statistical models to analyze and draw inferences from patterns in data with marginal human intervention. The procedure of Machine Learning is quite simple: it starts with inputting training data into the selected algorithm. Training data is used to develop the final Machine Learning Algorithm. To test whether this algorithm works correctly or not, new input data is fed into the Machine Learning algorithm. The prediction and results are then checked. If the prediction is not as good as it is expected the algorithm is re-trained many times until we get the preferred output and the accuracy of the prediction gradually increases over time. Here, the Python programming language,

its various libraries, and different Machine Learning algorithms will be used for different operations and output.

1.Logistic Regression

Logistic regression is a form of statistical approach to analyse data sets in which one or more independent factor impact the outcome. To examine the outcome, a dichotomous variable is utilized (where there are 2 outcomes). The purpose of it is construct the best fitting model utilized to represent the connection among a collection of independent variables (predictor or explanatory) and a diachotomous feature of interest (dependent variable= response or outcome variable). To predict the result of categorical dependent variable, a specific Machine Learning categorization approach called logistic regression is utilized. In a logistic regression's approach, the dependent variables are bi nary variables that comprises of data which is coded as one (yes, success, fair, etc.) or zero ('no, failure, etc.').

2.Naïve Bayes

The Naive Bayes method is supervised learning techniques for categorization sums that depends on Bayes theorem. It is basically, mostly utilized for text categorization problems that mostly need large training datasets. The Naive Bayes Classifier is simple yet powerful categorization way which assists in the creation of quick ML models that are able to make accurate prediction. The Naive Bayes Algorithm is especially utilized in spams filtering, sentiment observation, and articlesclassification.

3.Decision Trees

It's a popular, known as well as a powerful procedure. The algorithmsof decision trees is a standard learning procedure. This works with both categorical and continuous output variable. Decision-trees construct categorisation model in a format like a tree structure.

It divides datasets into small and further finer subsets over time whilst also building deciding trees. A categorisation is shown by leaf-node, and any decision nodes shall have two or more branches. The finest forecaster are shown by root-nodes, that is topmost resulting-node in a tree. The two, the categorical along with, the numeric data could be managed by decision-trees. As tree- structures, the decision-trees construct categorisation model. For categorisation, then it shall employ an if and then rule-set which is both unique and comprehensive. The regulations can be then memorized respectively, putting into use the training information. The tuples wrapped by the rules are removed when every turn a regulation is memorized.

On the training set, these processes reoccur till a certain end rule is met. It's formed in a repeating division and achieving style from the top to down. Every attribute must also be categorical. If not, they must be segregated well before of their schedule. The data receiving idea is then used to guess properties on topmost of trees which gives a bigger effect on categorisation. An increased number of branches can result from over fitting a decision tree to revealing anomalies due to noise or outliers.

4. K nearest neighbour

It is one of the easiest and most popularly utilized ML algorithms which is a form of Supervised Learning Algorithms. It predicts what is similar between the new case or data and the cases already present and put new case into that subset which has all the similar properties in the already present data set. It contains already present data and new data point is classified according to it’s similar nature. So, when there is appearance of fresh data it can be put into a category with the help of K- NN algorithm. K-Nearest Neighbor is the non parametric algorithm, and cannot make any ideas or assumptions on the data. It is also called ‘lazy-learner-algorithm’. It just stores a dataset and on getting the new data, that data is put into a same set as the new data.

1. AdaBoost

Boosting is ensemble modelling technique that was first observed and presented in the year 1997 by Freund and Schapire and it is since then that it has been a very utilized technique for managing binary categorization problems. These algorithms come handy in improving the prediction power by changing the weak learners into strong learners. It is best utilized to improve the efficiency of decision tree on the binary categorization problems.

More recently Adaboost is said as discrete AdaBoost because it can be utilized for the categorization purpose instead of the regressions. It is utilized to improve the efficiency of any machine learning algorithms. They are the samples that give the accuracy of random chance on any categorization problems. The widely utilized and well suied and hence most utilized algorithm in association with the AdaBoost are the decision-tree with only one level. Hence the tree is very small and they have one decision for categorization, called as decision stump.

2.Random Forests

Random Forests is widely known ML method which is the 'supervised learning' technique. It is utilized for either the categorization or Regression problems in Machine Learning. It was mainly derived from ensemble learning, a process of bringing together variousclassifiers which solves tough problems and hence to make the model efficient.

Random Forest is the type of classifier that has many decision trees on multiple sub sets of given a data set and then finds the mean to increase the predictive efficiency. Instead of only depending on one of the decision trees the random forest ensures prediction from the each of the trees present and seeing that the number of majority votes of predictions, the final output is predicted.

3.Support Vector Machine

It is very popularly and most utilized 'Supervised Learning' processes, utilized for the classifications and Regression problems. But, in majority of cases, it is utilized for classification problems in ML.

It's objective is to make the decision boundary which can differentiate between n dimensional space with classes so it becomes easier in inserting new data points in the right order in future. This is called a hyperplane.

Extreme points or vectors are chosen by Support Vector Machine that can be utilized in making hyperplane. Support vectors are extreme cases, and so it is termed as Support Vector Machines. Those vectors or data points nearest to the hyperplane and affects the position of hyperplane are actually termed as Support Vectors. Since hyperplane is supported by these they are called a Support vectors.

5. Kmeans clustering algorithm

Algorithm: Kmeans cluster K

Input : Dataset of size M x N, and No. of Cluster k

Output: k clusters of n objects

Step 1: Initialize Cluster centroid randomly.

Step 2: Calculate the distance between the data points and the cluster centroid by using dissimilarity measures. Depending upon the minimum distance, data points are partitioned into k clusters.

Step 3: Compute the new centroid for each and every cluster.

Step 4: Go to Step 2 and continue this procedure until cluster label does not change anymore.

Advantages:

1) Easy to implement and robust.

2) Relatively adaptable and productive in preparing enormous informational

indexes with direct time multifaceted nature.

3) Produce more tightly groups than various leveled bunching Produce tighter clusters than hierarchical clustering

Disadvantages:

1) Applied just when the mean of a bunch is characterized.

2) Cannot be applied on categorical attributes

3) Sensitive to the selection of number of a clusters k and initial cluster center.

4) Sensitive to noise and outlier data points.

We have gone through literature for finding no. of optimal cluster. Researchers suggest to use Elbow method, Average Silhouette index and gat stat index to find value of k cluster

So, we have applied these three methods on our intrusion dataset.

Find optimal k no. of clusters

1) Elbow Method

The main purpose of the elbow method is to run k-means clustering on the dataset for a range of values of k (say, k from 1 to 10), and for each value of k compute the sum of squared errors (SSE). Our goal is to select a small value of k that still has a low SSE.

Step 1: Calculate clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters.

Step 2: For every k, calculate the total within-cluster sum of square (wss).

Step 3: Plot the curve that is according to the number of clusters k. Step 4: The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate Number of clusters.

2) Silhouette Method

The silhouette value is a measure of how similar an object is to its own cluster

(cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

Step 1: For each data point i, let a(i) be the average dissimilarity of i with all data points within the same cluster.

Step 2: Let b(i) be a lowest average dissimilarity of i to any other cluster.

Step 3: The cluster with this lowest average dissimilarity is said to be the neighboring cluster of i because it is the next best fit cluster for point.

3) Gap Statistic

The gap statistic compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data. The estimate of the optimal clusters will be value that maximizes the gap statistic (i.e, that yields the largest gap statistic). This means that the clustering structure is far away from the random uniform distribution of points.

4) Dunn Index

These cluster validity indices have been introduced in paper [7]. If a data set contains well- separated clusters, the distances among the clusters are usually large and the diameters of the clusters are expected to be small [3]. Therefore larger Value means better cluster configuration. each of the trees present and seeing that the number of majority votes of predictions, the final output is predicted.

4.Datasets

To find anomalies using ML algorithms, we require huge harmless and harmful network-traffic for testing and training step. But due to privacy issues we cannot use real network traffic. To treat this issue, many datasets were created in last decades. Here we have given information about some of the popular datasets.

4.1DARPA 98

This data set was made by the 'MIT Lincoln Research Center', with a aim to make a trainings and a testing environment for network 'Intrusion Detection Systems'. The US Air Force's local's computer- network is made in this dataset. Aside from benign/normal networktraffic, the dataset has 38 attack types that can be grouped together and given attack names including User to Remote (U2R), Denial of Service (DoS), Probe, and Remote to Local (R2L)_.

DARPA 98 has faced a lot of scrutiny, notably because it doesn't reflect real world network traffic, it is now out of date, and also it excludes streams that can be called false positives (the harmless information delegated assault, misleading alarm). However, the DARPA98 dataset is still noteworthy and the reason for that is it was utilized as to create other datasets ,for example, NSL-KDD and KDD Cup 99.

4.2 KDD 99

The University of California created this dataset for use by interruption discovery frameworks. In this dataset information packages from the DARPA98 dataset are utilized. The dataset is isolated into two sections as training part and test part. By using the highlight extraction cycle, 21 attributes were created that may be

utilized by machine learning algorithms. The preparation area comprises of 4898431 furthermore, the test area contains 311029 information stream. KDD99 has thirty eight assault types. 14 of them are simply intended for the test segment and address obscure attack.

4.3 CAIDA

CAIDA is a institute which examines web data. The data set given by the offices of this association is referred by a similar names. This dataset was created by combining many lengthy periods of information stream recording over 'San Jose City'. CAIDA also includes a portion that showshourly 'DDoS attack'.

Information streams in the 'CAIDA dataset' are demonstrated simply by straightforward application and explicit assaults. As a result, the range of research is somewhat limited. Furthermore, information streams are not identified in this informational collection.

4.4 NSL-KDD

Albeit the KDD99 dataset made in 1999 was a generally excellent option in contrast to DARPA98, it was seen that there are an excessive number of redundancies in KDD99, and these reiterations influence the outcomes of the investigations performed with it and the presentation of the AI calculations. In expansion, in light of the fact that the size of 'KDD99 dataset' is excessively huge, specialists attempt to utilize part of information sets. Sadly, to limit the dataset, arbitrarily chose information from inside could not catch every one of the properties of the informational index.

4.5 ISCX 2012

All the previous datasets that have been utilized for detecting anomalies cannot be utilized for real world scenarios because they are outdated . To handle this issue ISCX 2012 was created using a 7 day internet stream.Because it was

constructed utilizing genuine gadgets, genuine ordinary and malevolent streams which include 'IMAP,FTP, HTTP, POP3, SMTP protocols1 and SSH' were made.

- Every information is marked.

- Data includes many varieties of attack which include Denial of administration, Brute Force SSH, Distributed Denial of Service and Infiltrating).
- The ISCX 2012 dataset doesn't have TLS/SSL traffic, which has undeniably the greater part of the present Internet traffic so this dataset is not adequate to use.

4.6 CICIDS 2017

This dataset incorporates of a 5-day data streams on an network made with the aid of PCs using all the updated OS like Windows, Ubuntu 12/16, Macintosh and Kali.

Flow Recording Day (Working Hours)	**pcap File size**	**Duration**	**CSV File Size**	**Attack Name**	**Flow Count**
Monday	10 GB	All Day	257 MB	No Attack	529918
Tuesday	10 GB	All Day	166 MB	FTP-Patator, SSH-Patator	445909
Wednesday	12 GB	All Day	272 MB	DoS Hulk, DoS GoldenEye, DoS slowloris, DoS Slowhttptest, Heartbleed	692703
Thursday	7.7GB	Morning	87.7 MB	Web Attacks (Brute Force, XSS, Sql Injection)	170366
		Afternoon	103 MB	Infiltration	288602
Friday	8.2GB	Morning	71.8 MB	Bot	192033
		Afternoon	92.7 MB	DDoS	225745
		Afternoon	97.1 MB	PortScan	286467

Table 1. Details of CICIDS2017 Dataset

The advantage of CICIDS2017 over other dataset is:

- The acquired data is this present reality data which was collected from a testbed comprising genuine PCs.
- The streams of Data was gathered from PCs having modern working framework. There is working framework variety ('Windows, Mac, and Linux') between attacker and the victim's PCs.
- This dataset has more protocols than previous datasets which includes HTTPS
- Then again, a few impediments of this dataset are the accompanying:
- Raw data records and handled data documents are exceptionallyenormous.
- Dissimilar to the NSL-KDD and KDD99 datasets, this dataset not provides separate file devoted to training and the testing.

  We have decided to use CICIDS2017 processed data as the dataset utilized in the implementation stage because of the comparison process. The datasetis updated and has wide range of protocols and attack pool.Therefore we choosethis dataset.

- Sets of the data are marked.

- Both raw data as well as processed data are accessible to work.

4.7 Research Approach

The primary task was to characterize the target network in terms of suitable network parameters. The parameters are chosen such that their values will change perceivably in normal and intrusive conditions. The features considered are the commonly seen protocols in the network traffic, the traffic data rate and the flow direction. In essence, the Anomaly model tries to capture the network behaviour in terms of two quantities intensity and heterogeneity. Intensity refers to the number of occurrences of a given network parameter over a period of time (for example number of TCP connections or number of outgoing HTTP packets etc) while

heterogeneity refers to the observed pattern of the nature of network activities over time (for example the data rate of HTTP packets in different time segments of the day or observations like web traffic is more during the beginning of office hours and then drops. It rises again during the closing hours etc). These two quantities closely relate to activities occurring in any given network and thus can represent the behaviour of network under the assumption that network behaviour has certain degree of repeatability. Once the network behaviour is quantified with these parameters, the next step would be to observe how they vary with time. The observation has to be made on different days of a week because the network behaviour changes over working days and non working days of a week and also on general holidays. The Anomaly based IDS has two operational modes.

Learning (or training) mode:

In this mode, the IDS learns the normal traffic behaviour in terms of representative feature set characterizing the target network. It collects the statistics of the selected network parameters for different types of days (Week days from Monday to Friday, Saturdays and Sundays) and then stores them into a specified file for subsequent processing. The frequency of statistics collection is set as per requirement; it is set by default to 10 minutes. IDS is put in this mode for sufficient period to learn the normal network behaviour. Sufficient training period is the key factor in reducing the false alarms. When IDS is learning the normal behaviour, the target network is assumed to be free from attacks and intrusions Following attributes are considered for characterizing the network:

TCP Packet count (incoming, outgoing and within LAN)

UDP Packet count (----------’ ’----------)

ICMP Traffic (----------’ ’----------)

The number of TCP connections

Web Traffic (incoming, outgoing)

DNS Traffic (---------’ ’----------)

Data rates TCP traffic in kb/s (---------' '----------)

Data rates UDP traffic in kb/s (---------' '----------)

Data rates HTTP traffic in kb/s (---------' '----------)

Data rates DNS traffic in kb/s (---------' '----------)

Once the learning is over, profile for the target network is generated with the gathered data using a profiler. If statistics collections is done at every 10 minutes and the learning period is say 1 month, total 24 sample values are available for each network parameter corresponding to each hour of the week day. Hence the profile is generated for each hour of the day over entire week. This implies that total 168 baseline vectors are established for the entire week, each vector containing 25 network parameters. The profile also contains 168 inverse matrices each of the order 25 x 25, accounting for number of parameters in consideration. This profile is used by Anomaly detection module during the detection phase. The IDS is also trained to learn the network behaviour in the presence of network intrusions. Intrusions are simulated using the MIT-DARPA training data set. Network profile
Is also generated for this condition.

When the network environment changes for genuine reasons, it may result into a number of false positives. In such situations the Anomaly model can be updated by rerunning the training phase on the changed traffic and rebuilding the profile using profiler program.

Input : The file containing the features values logged during the learning phase

Output : files containing the mean, standard deviations and inverse matrices of feature set

begin

for i =1 to Num. of week days do

for j =1 to Num. of hours in a day do

Read the feature values logged during learning phase;

for k =1 to Num. of network features do

find sum of the values corresponding to the same hour and day of the week;

Compute Average values and standard deviation for each feature;

Compute $\sum_{l,m=1}^{n}(x_1 - \mu)(x_m - \mu)^T$ *where n is the total number of features*

Compute the Determinant of above covariance matrices

if Determinant ≤ 0

Consider the neighbouring covariance matrix having positive Determinant

Compute inverse matrix corresponding to each Covariance matrix end

end

Algorithm for generating the profile

Detection mode:

In this mode, IDS detects in real time, the network based attacks leading to abnormal traffic pattern. The abnormality is decided on the basis of the network profile constructed earlier. The profile contains 168 vectors corresponding to each hour of the day over entire week, each vector containing as set of 25 features which describes the network. The Anomaly detection module samples the selected network parameters at regular intervals, as in the case of learning mode, checks whether they comply with already established network profile for that particular hour and day of the week. If it detects significant deviations, then it triggers alerts.

Input : The file containing the network profile

Output : Sends alert in case a event is detected as intrusion begin

Begin

for i =1 to Num.of week days do

for j =1 to Num. of hours in a day do

for k =1 to Num. of network features do

Read Average values and standard deviation for each feature;

Read the inverse matrices

Read the determinant matrix corresponding to each inverse matrix

Compute ($\mu \pm \sigma$) for each parameter

if ($\mu - \sigma > x > \mu + \sigma$)

x is intrusive

Compute $T^2 (X - \mu)S^{-1} (X - \mu)^T$

If T^2 *exceeds the threshold flag alerts*

Compute $g_i (X) = -\frac{1}{2} ln|S| - \frac{1}{2} (X - \mu)^T S^{-1}(X - \mu) + ln\, p(I)$

If $g_i (X)$ *exceeds the threshold flag alerts*

end

Algorithm for Anomaly based detection

5. Implementation

5.1Data Cleansing

We must make some necessary some changes to the dataset before using it in practice in order to make it more accurate as well as efficient. Some errors of the CICIDS2017 dataset are fixed in this segment, and some data is edited for this purpose. This dataset has 3119345 stream records. Distribution of the records is shown in Table below. On comparing the records, it becomes clear that record 288602 is either incorrect or incomplete. our 1st step should be to remove the no-longer-needed data.

Label Name	Number
Benign	2359289
Faulty	288602
DoS Hulk	231073
PortScan	158930
DDoS	41835
DoS GoldenEye	10293
FTP-Patator	7938
SSH-Patator	5897
DoS slowloris	5796
DoS Slowhttptest	5499
Bot	1966
Web Attack – Brute Force	1507
Web Attack – XSS	652
Infiltration	36
Web Attack – SQL Injection	21
Heartbleed	11

Table 2. Distribution of stream records in the CICIDS2017 dataset.

..

Another flaw in the utilized dataset can be found in the segments that make up the features The Flow ID, Source Port, and Source IP are among the 86 segments in the dataset record that describe the stream's properties. On the other hand, the Fwd Header Length feature was written twice (in the 41st and 62nd columns). To correct this error, the rehashing segment is removed (column 62).

Some features cannot be used directly and should be changed in order to work with for example the string values need to be changed tomathematical data. Sklearn classes' LabelEncoder() should be able to dothis. Nevertheless the "Label" tag is unchanged because we require such category to identify attacks and to implement different methodologies.

At long last, a few minor underlying changes ought to be made are:

- In the Label feature, the character"- " utilized in distinguishing the attack subtypes is changed with "- ".
- "Flow Bytes/s", "Stream Packets/s" features incorporate the qualities "Infinity" as well as "NaN" in expansion with mathematical qualities, which is altered to -1 and 0 separately.

5.2 Data Preparation

Data is required for ML algorithms in order for learning to take place.We require data for training as well as to evaluate the performance of algorithm. The computation applies what it has learned from the training data to the test data.The effectiveness of the ML algorithm is determined by test data.

CICIDS2017 dataset, on other hand, doesn't contain distinct data for training and testing, except a single unbundled dataset. So we have to divide the dataset ourselves.We use a Sklearn command, train test split, for this purpose. It will divide the data into two sections as specified. The majority of the time, we divide the dataset in 20 percent test and 80 percent train data.It assures that the choice is

made at random. To ensure that outcomes obtained are accurate, the previous step has been performed several times.

5.3 Feature Selection

The properties of this dataset are evaluated to figure out the relevant features to identify each attack.

Selection of Features Based on the Types of Attacks

We must develop a file for each sort of attack, segregating the attack from other attacks, in order to perform this computation.It contains the whole stream labelled "attack" as well as data stream labelled "Benign" at random.

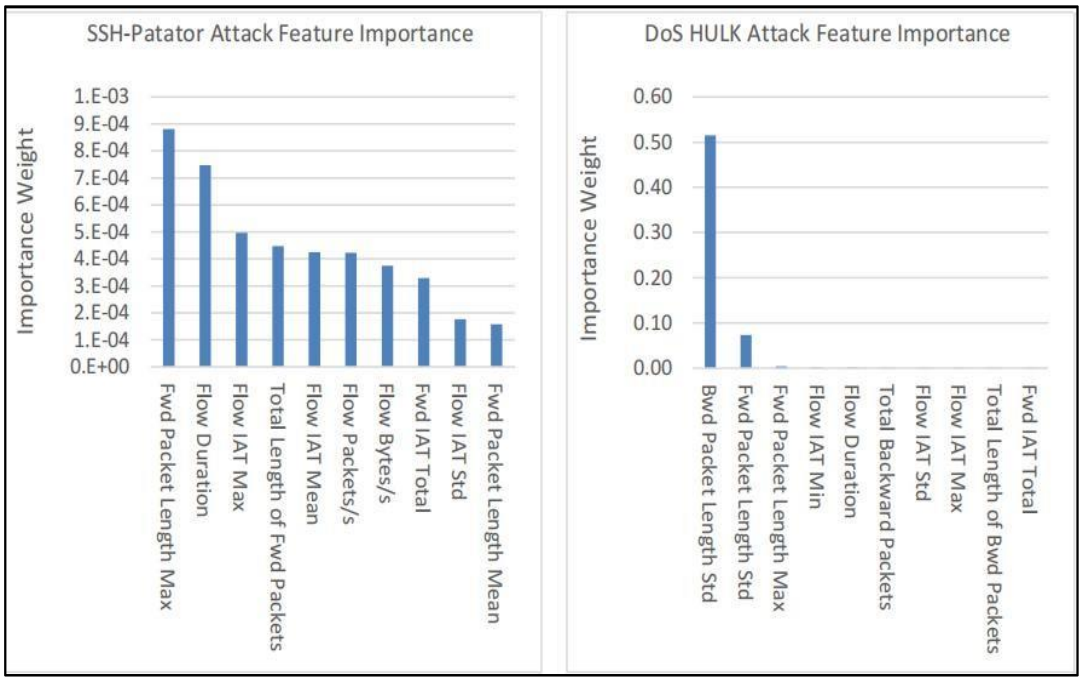
We use Sklearn's Random Forest Regressor class to determine importance of properties. As a result of this calculation, a decision- forest is created. Each feature inside the decision forest is assigned a weighted importance based on its then sorted at the end of the cycle. The sum of the importance loads of a large number of properties determines the decision tree's aggregate importance weight. The relationship between the score of any feature and the overall tree score provides information about the feature's importance in the decision tree.

While determining weight of significance, 8 properties (“Flow ID, Source IP, Source Port, Destination IP, Destination Port, Convention, Timestamp, External IP”) should be excluded from the computation. Despite the fact that these properties are used in traditional ways, it's possible that the hacker wouldn't utilize notable ports to get away.

It will be far more successful to exclude deceptive properties for example “IP address, Port number, and Timestamp” when defining the

attack's attribute significance. Instead, utilise more general and invariant attributes to characterize the attack.

The "Heartbleed" and "SSH-Patator" attacks have a wide range of characteristics. There are several properties of these two attacks that have important results that are comparable to one another. Reliable defenses to these attacks are an unsolved challenge. In this work, we present a novel evasion attack: the 'Feature Importance Guided Attack' (FIGA) which generates adversarial evasion samples. FIGA is model agnostic, it assumes no prior knowledge of the defending model's learning algorithm, but does assume knowledge of the feature representation. FIGA leverages feature importance rankings; it perturbs the most important features of the input in the direction of the target class we wish to mimic



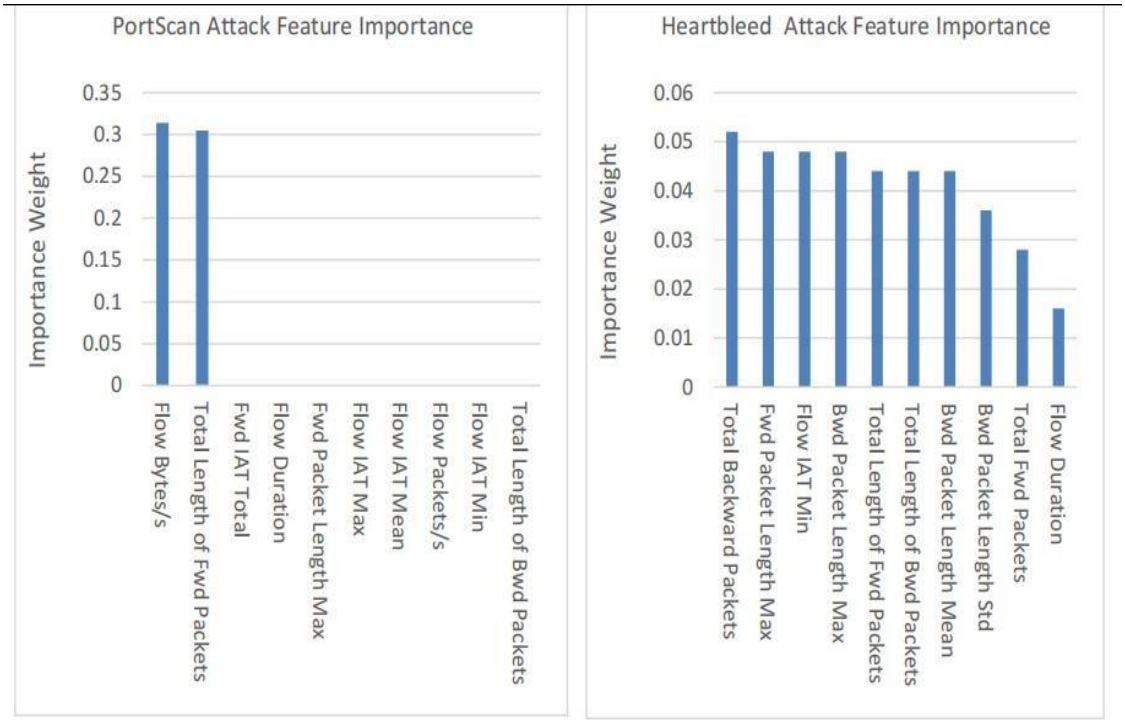


Figure 13. Graphs of feature importance weights of SSH-Patator, Heartbleed, DoS HULK, and PortScan attacks

In the PortScan attacks, the two features which are “Flow Bytes/s “and “Total Length of Forward Packets” stick out. When analyzing PortScan attack, the attacker often tries to transmit many packets as feasible with no or little payloads.

As a result, the attacker speeds up the process while simultaneously increasing the efficiency of the attack and effectively managing thebandwidth.

The importance values of the SSH-Patator are extremely close together. The reason being “SSH-Patator” attack is a complex three-stage attack (Scanning Stage, Brute-Force Phase, and Die-off Phase).

Feature Selection According to Attack or Benign

By merging all attack kinds under a single category: "attack," we performed Random Forest Regressor on the full dataset. Now this file only contains attack and benign labels.The Table below shows the feature list collected, and Fig 16 shows the features graphic.

Feature Name	Importance Weight		Feature Name	Importance Weight
Bwd Packet Length Std	0.246627		Flow IAT Mean	0.003266
Flow Bytes/s	0.178777		Total Length of Bwd Packets	0.001305
Total Length of Fwd Packets	0.102417		Fwd Packet Length Min	0.000670
Fwd Packet Length Std	0.063889		Bwd Packet Length Mean	0.000582
Flow IAT Std	0.009898		Flow Packets/s	0.000541
Flow IAT Min	0.006946		Fwd Packet Length Mean	0.000526
Fwd IAT Total	0.005121		Total Backward Packets	0.000169
Flow Duration	0.004150		Total Fwd Packets	0.000138
Bwd Packet Length Max	0.004007		Fwd Packet Length Max	0.000125
Flow IAT Max	0.003579		Bwd Packet Length Min	0.000084

Table 4. According Attack and Benign Labels Feature Importance Weight List

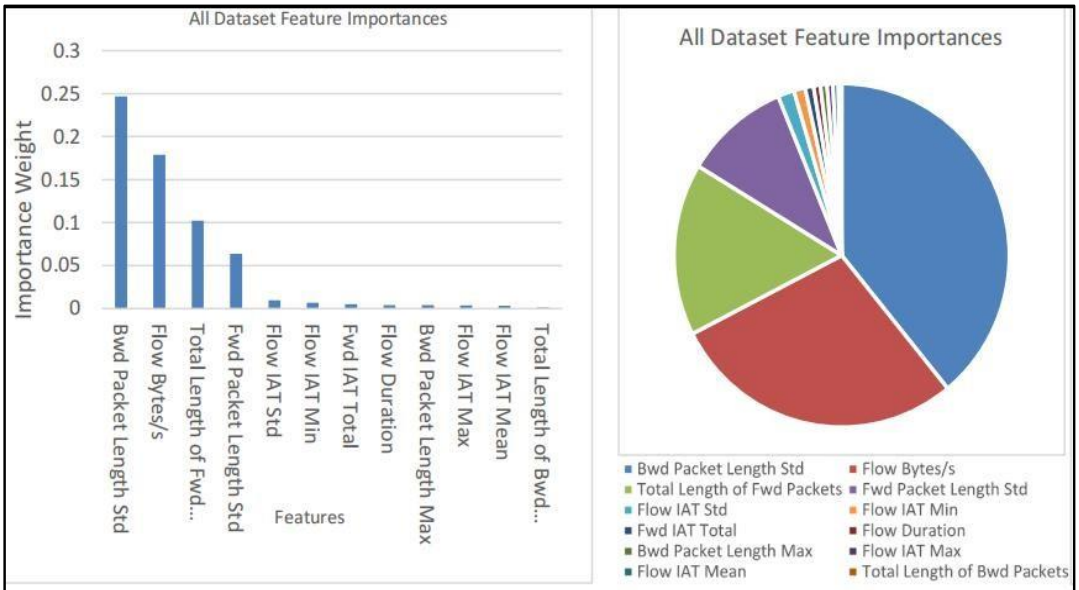


Figure 16. Graphs of Feature Importance Weights According to Attack and Benign Labels

6. Implementation of Machine Learning Algorithms

We applied ML methods to the dataset using 2 different methodologies. The file obtained in the first method ie Feature Selection area, and the properties obtained in a similar area, are utilized in the main method. These file has 30% attack data and 70% benign data, and they are labeled with the type of attack they contain. The 7 ML algorithms are applied on each record multiple times, yielding different results for each attack type. Using this method, we can observe the effectiveness of different algorithms.

The entire dataset is utilized as a single document in the 2nd method. Allof the attacks in this document are grouped together under one category: "attack." The data now has only attack and benign labels. We are using the top four properties having highest importance-weight for all attacks. As a result, four properties are selected from the twelve attacks, making a total 48 attributes. After removing redundancies, total features is reduced to 18. Table 5 should show a list of these features.

Table 5

Bwd Packet Length Max	Flow IAT Mean	Fwd Packet Length Min
Bwd Packet Length Mean	Flow IAT Min	Fwd Packet Length Std
Bwd Packet Length Std	Flow IAT Std	Total Backward Packets
Flow Bytes/s	Fwd IAT Total	Total Fwd Packets
Flow Duration	Fwd Packet Length Max	Total Length of Bwd Packets
Flow IAT Max	Fwd Packet Length Mean	Total Length of Fwd Packets

This method can be implemented in an alternate way by using properties having high weight based on importance scores derived for the full data therefore not using 18 features above.

The feature weight Cutoff value was set at 0.8 percent. In this method, only 7 attributes will cover 97 percent of the overall feature important weight. The other 13 properties account for barely 3% of the overall weight of importance. The below properties are used if the properties with a weight of 0.8 percent and higher are selected:

Feature Name	Importance Weight	Percentage
Bwd Packet Length Std	0.246627	38.97%
Flow Bytes/s	0.178777	28.25%
Total Length of Fwd Packets	0.102417	16.18%
Fwd Packet Length Std	0.063889	10.10%
Flow IAT Std	0.009898	1.56%
Flow IAT Min	0.006946	1.10%
Fwd IAT Total	0.005121	0.8 %

6.1 Using 12 Attack Types

7 different ML algorithms were applied to twelve attacks and theoutcome can be seen in Table below.

Attack Names	F-Measures						
	NB	RF	KNN	ID3	AB	MLP	QDA
Bot	*0.54*	0.96	0.95	0.96	**0.97**	0.64	0.68
DDoS	0.77	0.96	0.92	**0.96**	0.96	0.76	*0.34*
DoS GoldenEye	0.81	0.99	0.98	**0.99**	0.99	*0.64*	0.71
DoS Hulk	*0.23*	0.93	0.96	**0.96**	0.96	0.95	0.36
DoS Slowhttptest	*0.35*	0.98	0.99	0.98	**0.99**	0.78	0.38
DoS slowloris	*0.37*	0.95	0.95	**0.96**	0.95	0.74	0.46
FTP-Patator	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00
Heartbleed	**1.00**	0.99	1.00	0.95	0.93	*0.66*	1.00
Infiltration	0.78	0.92	0.88	0.89	**0.92**	*0.52*	0.83
PortScan	*0.39*	1.00	1.00	**1.00**	1.00	0.61	0.85
SSH-Patator	*0.33*	0.96	0.95	**0.96**	0.96	0.83	0.41
Web Attack	0.74	0.97	0.93	**0.97**	0.97	*0.60*	0.84

Table 7. Distribution of results according to type of attack and machine learning algorithm.

By looking at the results we can observe that these algorithms (“RF, KNN, ID3 and Adaboost “), have made more than 90 percent progress inidentifying the attacks . Out of these 4 methods, ID3, has finished 7 of twelve responsibilities with the most noteworthy score. As a matter of fact, in the six of these seven attack types (“DDoS ,DoS Hulk, DoS GoldenEye,Web Attack, PortScan and SSHPatator”) ID3 gives best performance with atleast 1 algorithm. In any case, low process timing makes it stand out in front of all rest of the methods.

NB, has least F measure and is the last in 50% of the tasks. QDA has an exceptionally close score to Naive Bayes. Naive Bayes has lower performance than the other algorithms. One more intriguing point about NB algorithms is it has the most elevated score in case of FTPPatator. Although NB has poor performance when compared to other methods but it is much better in terms of speed.

Almost all the ML algorithms accomplish a complete score in the caseof FTP-Patator. The reason is because of the way that the properties utilized to portray FTPPatator might turn into a trademark that is effectively recognized as typical and attack data. MLP has the second- worst performance after Naive Bayes. In four of the twelve attacks, it has been the last one.

6.2 Using Two Groups: Attack and Benign

Here we have utilized the complete dataset as a single file. This record's attack are grouped together under a single common label, "attack." This dataset is subjected to 7 different ML methods. Two approaches will be utilized: the first 2nd make use of the features developed for attacks file in 1st approach. The second approach utilizes the seven properties collected in “Feature Selection According to Attack or Benignsection”.

Reliable defenses to these attacks are an unsolved challenge. In this work, we present a novel evasion attack: the 'Feature Importance Guided Attack' (FIGA) which generates adversarial evasion samples. FIGA is model agnostic, it assumes no prior knowledge of the defending model's learning algorithm, but does assume knowledge of the feature representation. FIGA leverages feature importance rankings; it perturbs the most important features of the input in the direction.

6.2.1Using Features Extracted for Attacks Files

The table below shows outcome achieved utilizing the 18 properties from the 1st approach. The best performing algorithm is KNN having a score of 0.96. Next is ID3 and Adaboost with a score of 0.95. Also ID3 is remarkably fast as compared to Adaboost. The worst score is obtained by QDA having a score of 0.30. Naive Bayes and QDA are the fastest algorithms. Also, we can observe that though KNN has the best score itis the slowest algorithm.

Machine Learning Algorithms	**Evaluation Criteria**				
	F-Measure	Precision	Recall	Accuracy	Time[5]
Naive Bayes	0.79	0.80	0.78	0.78	**4.576**
QDA	*0.30*	0.84	0.31	0.31	6.649
Random Forest	0.94	0.95	0.94	0.94	24.739
ID3	0.95	0.95	0.95	0.95	29.284
AdaBoost	0.95	0.95	0.95	0.95	391.804
MLP	0.79	0.81	0.84	0.84	81.668
K Nearest Neighbours	**0.96**	0.96	0.97	0.97	*1967.054*

Table 8. Application of the features obtained in the first approach.

6.2.2Using Feature Selection for complete Dataset

Table below shows an implementation that utilizes a different feature selection method. The seven features achieved in the "FeatureSelection According to Attack or Benign" area are utilized here. The parts that are changed is colored in red. When we compare table 9 and table 8 there is no change in MLP, AdaBoost, ID3and Random Forest .But in the case of QDA and Naive Bayes there is an increase of 11 and 2 points respectively. If we see the time then we can notice that all the algorithms are now faster because of the reduction in the number of features taken.

Machine Learning Algorithms	Evaluation Criteria				
	F-Measure	Precision	Recall	Accuracy	Time
Naive Bayes	0.81	0.8	0.82	0.82	1.6258
QDA	0.41	0.83	0.38	0.38	1.925
Random Forest	0.94	0.947	0.94	0.94	20.511
ID3	0.95	0.95	0.95	0.95	11.552
AdaBoost	0.94	0.94	0.94	0.94	144.166
MLP	0.79	0.815	0.84	0.84	51.799
K Nearest Neighbours	0.97	0.97	0.97	0.97	1038.253

Table 9. Implementation of features obtained using Random Forest Regressor for All Dataset.

6.3 Experimental Results And Discussion

To evaluate the system, two major indicators of performances are chosen.

- Detection rate
- False positive rate

Detection rate is defined as the number of intrusion instances detected by the system divided by the total number of intrusion instances present in the test set. The false positive rate is defined as the total number of instances that were wrongly detected as intrusions divided by the total number of normal instances. These are good measures of performances since they measure what percentage of intrusions the system is able to detect and how many incorrect classifications it makes in the process. The following sub sections give the details of evaluation scheme and the Results obtained.

6.4 Evaluation Scheme

The Anomaly IDS is trained for five weeks to learn the normal network traffic of the IIT, Kharagpur. The model considers a vector of 25 network attributes to describe the target network. The IDS is also trained for more than three weeks to learn the network behaviour under intrusions. The intrusions are simulated in the network using MIT-DARPA 1999 data set. The training data contains a total of 4396 vector data points for normal traffic and 2120 vector data points for intrusive traffic. The training period covers different types week days (working, Saturday and non working days). The network profile is generated using the training data which contains a total of 168 vector data points corresponding to each hour of the day over the entire week. The same training data and the test data is used with all the three techniques discussed earlier.

About MIT-DARPA IDS Evaluation

In 1998, the Information Systems Technology Group of Lincoln Laboratory at MIT, in conjunction with the Air Force Research Laboratory (AFRL) and the Defence Advanced Research Projects Agency (DARPA), began work to develop a standard for the evaluation of Network IDS. Developing this evaluation meant the creation of consistent and repeatable network traffic. The traffic was created through the study of 4 months of data from Hanscom Air Force Base and approximately 50 other bases. Using that data, they were able to generate and simulate network traffic, while introducing attacks, probes and intrusions into the data. Both training and testing data were simulated and two types of traffic were published. Training data is traffic in which the attacks were known from the start. A second set of data contains traffic in which the attacks were not described explicitly. Data sets of Week 1 and Week 3 contain attack free traffic while Week 2 contains training data with attacks. Week 4 and Week 5 are the testing data containing network attacks in the midst of normal background data. Test Data sets contains four categories of simulated attacks DoS – Denial of service (e.g. SYN flood) R2L -- unauthorized access from remote machine (password guessing) U2R –unauthorized access to super user or root functions (buffer overflow attacks) Probing --surveillance and other probing vulnerabilities (port scanning) A more complete discussion on this is available at the Lincoln Laboratory/ MIT site Detection rate is defined as the number of intrusion instances detected by the system divided by the total number of intrusion instances present in the test set. The false positive rate is defined as the total number of instances that were wrongly detected as intrusions divided by the total number of normal instances. These are good measures of performances since they measure what percentage of intrusions the system is able to detect and how many incorrect classifications it makes in the process.

7.Conclusion

Network Intrusion Detection System has a major role to play in safeguarding the network resources against various kinds of attacks. With the advent of new vulnerabilities and sophistications in the nature of attacks, new techniques for intrusion detection have evolved. The main objectives of the research being increasing the detection accuracy while keeping the false positive rate low.

As stated earlier, the signature based techniques are good but has the obvious short comings like failure to detect novel attacks, increasing signature database etc. So the viable alternative would be to analyse the behaviour of the network as a whole and trying to build the model based on the observations. So Anomaly based detection has been a wide area of interest for researchers since it provides the base line for developing promising techniques.

The Anomaly based detection complements the Signature based technique and helps in identifying the novel attacks which lead to the anomalies in the network traffic. The major concerns in this method are identifying the appropriate network features to characterize the network and build a behavioural model and also the rate of false positives may increase sharply if the IDS is not trained sufficiently in the target network.

In the present framework of project, discussed the design and development of "Anomaly based intrusion Detection system" which is built on top of a existing open source signature based network IDS, called SNORT so to have both the analysis techniques in a single package.

The Anomaly based component of IDS is trained in the Computer and Informatics Centre of Indian Institute of Technology (IIT), Kharagpur where the IIT network traffic is sniffed using a port mirrored switch at the gateway. The IDS is trained for more than a month in the IIT network at computer and Informatics centre, to learn the normal traffic pattern. Also it is exposed to the intrusive traffic for more than 3 weeks, in a simulated environment by replaying the MIT DARPA

Intrusion Detection System training datasets (1999).

The thesis presented three techniques for detecting anomaly based intrusions at the network level. Statistical based anomaly detection techniques use statistical properties and statistical tests to determine whether "observed behaviour" deviate significantly from the "expected behaviour". The first technique is based on univariate statistic model with mean and variance. The second method uses the multivariate Hotelling's method while the last technique uses the Bayesian classification technique for discriminating attacks from that of normal activities.

All the three techniques are evaluated with the DARPA IDS evaluation Data sets (1999) and the results are compared. Bayesian approach proved to be a better solution than the Hotelling's Multivariate technique and the method of Statistical Moments.

Presently, the work caters only to identify and classify the events into normal and attack classes. It can be extended to detect and classify the attacks into multiple attack classes. Dynamic updation of the Anomaly Model using Bayesian Network can also be considered for future enhancement. Different Analysis techniques like HMM and Fuzzy Logic can also be tried as alternative techniques
for anomaly detection.

8. Future Work

The training and test data came from a set of CSV files containing features extracted from the network flow. Unfortunately, in real-world systems, this method is not feasible. If we record network data such that ML methods can be implemented on it, this problem can besolved.

It's also worth noting that different machine learning methods were utilized independently of one another, yielding experimental results. In practise, however, this method has limited applicability. A hierarchical ML system can be constructed to solve this challenge. Furthermore, a structure like this saves time, CPU power, and memory.

All environments are typically in a constant state of change, effective anomaly detection algorithms must continuously learn, rather than rely on heuristic models. If they don’t continuously learn, they risk missing subtle parameter changes, or can produce false positives.

Like any kind of machine-learning-based approach, the more data that is made available to the system, the more accurate the system becomes. This is one of the most important aspects of anomaly detection in general. The largest risks that your organization will encounter will almost certainly be unknown, or in other words, something you didn’t prepare for in advance. You need your system to be as experienced and accurate as possible in order to detect a new unknown that signals the risk. Using anomaly detection can prevent, or at the very least, catch new or unknown issues before they negatively impact your business is already connected.

9.References

- .C. F. Tsai, Y. F. Hsu, C. Y. Lin, and W. Y. Lin, “Intrusion Detection by Machine Learning: A Review ,” Expert Systems with Applications, vol.36, no. 10, pp. 11994 – 12000, 2009.

- R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, “Self-taught Learning: Transfer Learning from Unlabeled Data,” in Proceedings of the 24th International Conference on Machine Learning, ICML ’07, (New York, NY, USA), pp. 759–766, 2007.

- M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien, “Hybrid Intelligent Intrusion Detection Scheme,” in Soft computing in industrial applications, pp. 293–303, Springer, 2011.

- H. S. Chae, B. O. Jo, S. H. Choi, and T. K. Park, “Feature Selection for Intrusion Detection using NSL-KDD,” Recent Advances in Computer Science, pp. 184–187, 2013.

  Kumar S. Viinikainen A. & Hamalainen T. Machine learning categorization model for network based intrusion detection system, 2016 11th International Conference for Internet Technology andSecured Transactions (ICITST), 5-7 Dec. 2016. IEEE, Barcelona, Spain; Dec 2016. P 242−49.

  https://ieeexplore.ieee.org/document/7856705/ citations#citations.

- Jamal Esmaily, Reza Moradinezhad & Jamal Ghasemi. Intrusion Detection System Based on Multi-Layer Perceptron Neural Networks and Decision Tree. 2015 7th Conference on Information and Knowledge Technology (IKT), 05 October 2015. IEEE, Urmia: Iran; 2015. https:// doi.org/10.1109/IKT.2015.7288736.

- J. P. Anderson, Computer Security Threat Monitoring and Surveillance, Technical Report, James Anderson Report, Pennsylvania, (1980)

- Arif Jamal Malik, Waseem Shahzad and Farrukh Aslam Khan, Network Intrusion Detection Using Hybrid Binary PSO and Random Forests Algorithm, Security and Communication Networks, (2012).

- R. S. Naoum, N. A. Abid, and Z. N. Al-Sultani, "An Enhanced Resilient Backpropagation Artificial Neural Network for Intrusion Detection System," International Journal of Computer Science and Network Security, vol. 12, no. 3, pp. 11–16, 2012

- C. R. Pereira, R. Y. Nakamura, K. A. Costa, and J. P. Papa, "An Optimum-Path Forest Framework for Intrusion Detection in Computer Networks," Engineering Applications of Artificial Intelligence, vol. 25, no. 6, pp. 1226–1234, 2012.