MACHINE LEARNING BASED INTRUSION DETECTION SYSTEM USING STATISTICAL FEATURE RANKING METHOD

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY
IN
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

AKHIL KUMAR

2K20/CSE/501

UNDER THE SUPERSVISION OF

Dr. SHAILENDER KUMAR

(Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

May, 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Akhil Kumar, Roll No. 2K20/CSE/501 student of M. Tech (Computer Science and Engineering), hereby declare that the Project Dissertation titled "**Machine Learning based Intrusion Detection System using Statistical Feature Ranking Method Techniques**." which is being submitted by me to the Department of Computer Science & Engineering, Delhi Technological University, Delhi, in partial fulfilment of requirements for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi **Akhil Kumar**

Date: **2K20/CSE/501**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I, hereby certify that the Project titled "**Machine Learning based Intrusion Detection System using Statistical Feature Ranking Method Techniques.",** which is submitted by Akhil Kumar, Roll No. 2K20/CSE/501, Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date:

DR. SHAILENDER KUMAR
(Professor)

SUPERVISOR

Abstract

Big data has made it easier for people to live an information-based Internet lifestyle, but it has also created a number of serious network security issues that make it difficult to use networks on a regular basis. At the moment, intrusion detection systems are mostly used to identify aberrant network traffic. To keep track of packets entering a network, an IDS employs sensors. To find malicious packets, the packet data with the attack signatures it has stored in memory, and then compare the results. Another sort of IDS analyses the patterns of the monitored packets to spot packets that are attempting to attack the network. These IDSs are believed to be able to identify new sorts of assaults and detect packet irregularities. Both varieties of IDSs provide reports of malicious activities at the management console. An IDS offers an automated system to find both internal and external intruders. Firewalls are used to show and/or restrict the ports and IP addresses used for communication between two entities, whereas IDS are able to inspect the content of the packets before acting.The actual process of the current traffic incursion detection systems needs to be changed, nonetheless, due to their numerous flaws and high resource occupation rate. So, utilising a Machine Learning (ML) technique, we suggested a statistical analysis-based intrusion detection system in this study. In this paper, we suggested a mechanism for detecting intrusions by applying the T test, a statistical tool for ranking analysis: two sample assuming unequal variances. A substantial amount of network traffic data that includes both malware data and normal traffic data is gathered in order to identify the pattern of the malware data. The t-test is used to score nine different traffic aspects for

both intrusion and regular traffic, resulting in nine "t" values from which other features were deduced. The Naive Bayes machine learning algorithm will then be applied to the 9 features, deleting one feature at a time that has the lowest “t” value to provide 9 alternative accuracy values. After examining the accuracy value, we get to the conclusion that the two features with the lowest value are removed in order to attain the highest accuracy, with accuracy of the data increasing as each of those two lower features are removed. The accuracy percentage of our work is 95.69% achieved on top 7 features rather than using all 9 features. Hence, we can argue that feature ranking using T-test helps us in improving the overall detection accuracy.

Acknowledgement

This work would not have been possible without the constant support, guidance, and assistance of my major project supervisor Prof. Shailender Kumar, Dept. of Computer Science, DTU. Their level of patience, knowledge, and ingenuity is something I will always keep aspiring to.

Contents

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

S. No.	Abbreviation	Word/phrase
1	ML	Machine Learning
2	IDS	Intrusion Detection Systems
3	IDPS	Intrusion Detection and Prevention System
4	DDoS	Distributed Denial of Service
5	EFC	Energy-based Flow Classifier
6	RSU	Road Side Unit
7	CNN	Convolutional Neural Network
8	NB	NAÏVE BAYES
9	IPS	Intrusion Prevention System
10	HIDS	Host Intrusion Detection Systems
11	NIDS	Network Intrusion Detection Systems
12	IP	Internet Protocol
13	RePO	Reconstruction from Partial Observation

Chapter 1: Introduction

The rate of technological advancement is amazing. According to internet usage figures for the year 2017, 3.8 billion people, or half of the world's population, are currently online. Due to the vast array of applications it offers, internet usage is increasing, along with e-commerce, e-banking, email, and online shopping. Internet technology is like a two-edged sword; while it offers many advantages, it also raises concerns about privacy, data integrity, and accessibility. Computer network security is characterised as a process to protect the network from vulnerabilities in order to maintain its integrity and accessibility. To protect the network from numerous threats known as incursions is the aim of the network security technique and to stop them from entering the network and proliferating there as well. An intrusion is a group of actions that go against a computer network system's security protocol.

Network traffic is monitored by a security system called an intrusion detection system (IDS) on computers, analyses the traffic, and generates alerts or alarms when anomalies are discovered. IDS is described as a classifier that gathers information about whether an attack has taken place or not. IDS frequently recognises observations of an intrusion while it is occurring or by studying the results after an intrusion has occurred to detect intrusion/attack. IDS can analyse the packets moving in and out of the network and distinguish between normal and abnormal packets. Computer risks are identified via IDS, which also alerts the network administrator to a security flaw. IDS is used to enforce the complex security policies for network users, not to replace any of the current

security solutions on the network. IDS is therefore a defense-in-depth method that works in conjunction with firewalls and virus defence systems. Because of the device's inexpensive price, straightforward implementation, real-time detection, and prompt reaction, intrusion detection systems are attracting the attention of the network security sector. Sensors and a management console are used by the intrusion detection system. By comparing packet information with previously collected attack signatures, sensors find malicious behaviour and notify it to the management console. Information security officers, database managers, and network administrators are rapidly expanding their usage of IDS technology because it offers an automated method of detecting both internal and external intruders.

Comparing an intrusion detection system to a standard firewall system reveals how much better it is. IDS can display the content of packets, but firewalls can only display the IP addresses and ports utilised for inter-entity communication. Sensors in an intrusion detection system are able to identify nefarious behaviour because they are familiar with how the protocols work. IDS has several benefits over conventional security technologies, however the technology still has some drawbacks. An intrusion detection system will typically produce a deluge of alarms. The alert may be a false positive that overburdens the system's ability to process data, therefore it does not necessarily indicate malicious behaviour. Since 99% of notifications from intrusion detection systems are false positives, human analysis of the alerts is necessary. Since the system for detecting intrusions occasionally drops packets when the network is overloaded with

large amounts of data, it looks to be unstable. As a result, there is a higher chance of missing actual intrusions. A secure network is necessary to protect sensitive data from both internal and external intruders due to the increase in internet usage. Intruders who are already on the network and have access to all of its important data pose the biggest threat to the network security plan. Firewall technologies are quite successful at keeping networks safe from unauthorised outside access. The system for detecting intrusions keeps track of the data entering and leaving the network, examines the data, and alerts the user when an abnormality is found. The comparison makes it simple to understand the differences between firewalls and IDS. Let's imagine you keep pricey stuff in your house for storage. Installing home security equipment, such as closed-circuit video cameras, and establishing barriers like gates are two ways to secure this asset. While firewalls might be equated to closed gates, IDS are the CCTV cameras or security systems. Filtering and stopping anomalous network traffic is what firewalls do. IDS handle the procedures of sniffing, analysing, and alerting. Finding computer intrusions and alerting the network administrator to the security failure are the main responsibilities of IDS. IDS is intended to enforce the complex security policies for network users rather than replace any of the current network security solutions. Figure 1 demonstrates the installation of an intrusion detection system and a firewall on the network. Protecting network confidentiality, integrity, and assurance from various intrusions is the main objective of system for detecting intrusions. A system for detecting intrusions is more of a reactive than a proactive informant for a network administrator. IDS cannot halt invasions; it can only identify them. Intrusion detection systems are also capable of studying audit data to

comprehend the behaviour and impact of intrusion, which can be used to construct more complicated IDS systems.

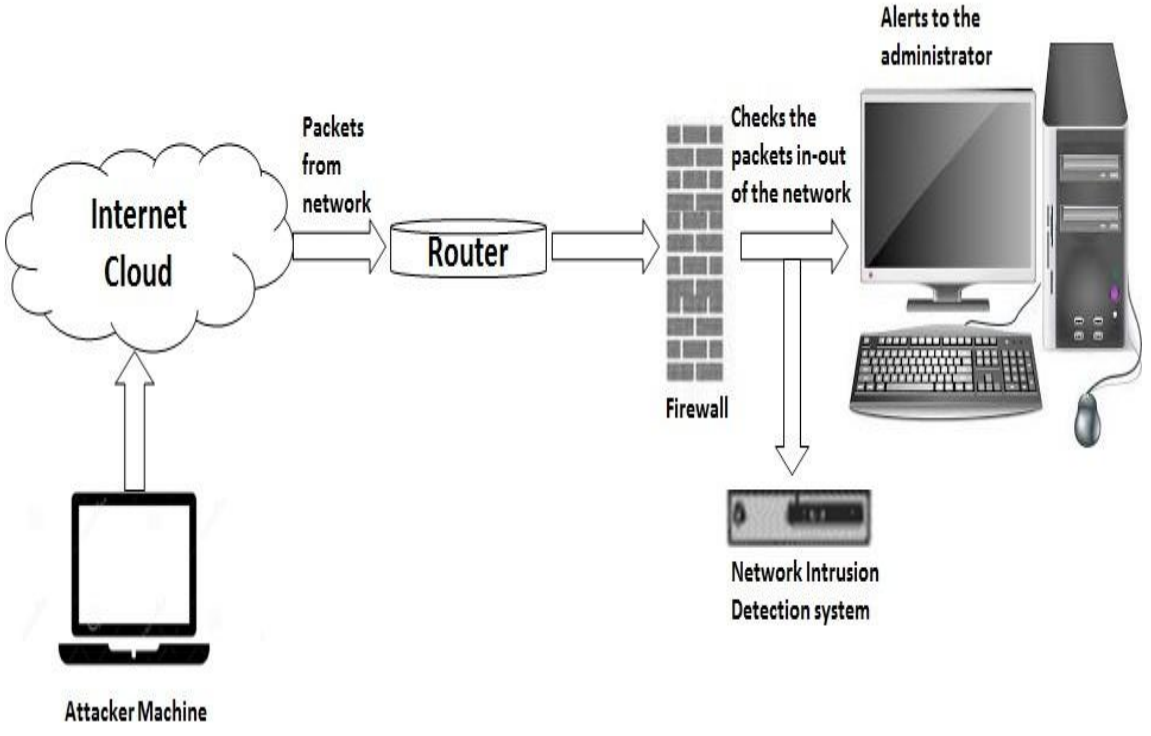

Figure 1: Deployment of Firewall & Intrusion Detection System

The two primary components of an system for detecting intrusions are the control console and the sensors as shown in figure 2. The management console includes a knowledge database for attacks information about the system's current state, audit information outlining occurrences, and a response engine that manages the reaction mechanism and how to respond. Sniffing network

traffic and assessing audit patterns are tasks that sensors carry out. The system has two choices: block the source of the assault or sound an alarm and alert the administrator. A reliable intrusion detection system should be capable of running continuously without human intervention. It should be better equipped to identify attacks and erroneous alerts and function with the minimal amount of system overhead.

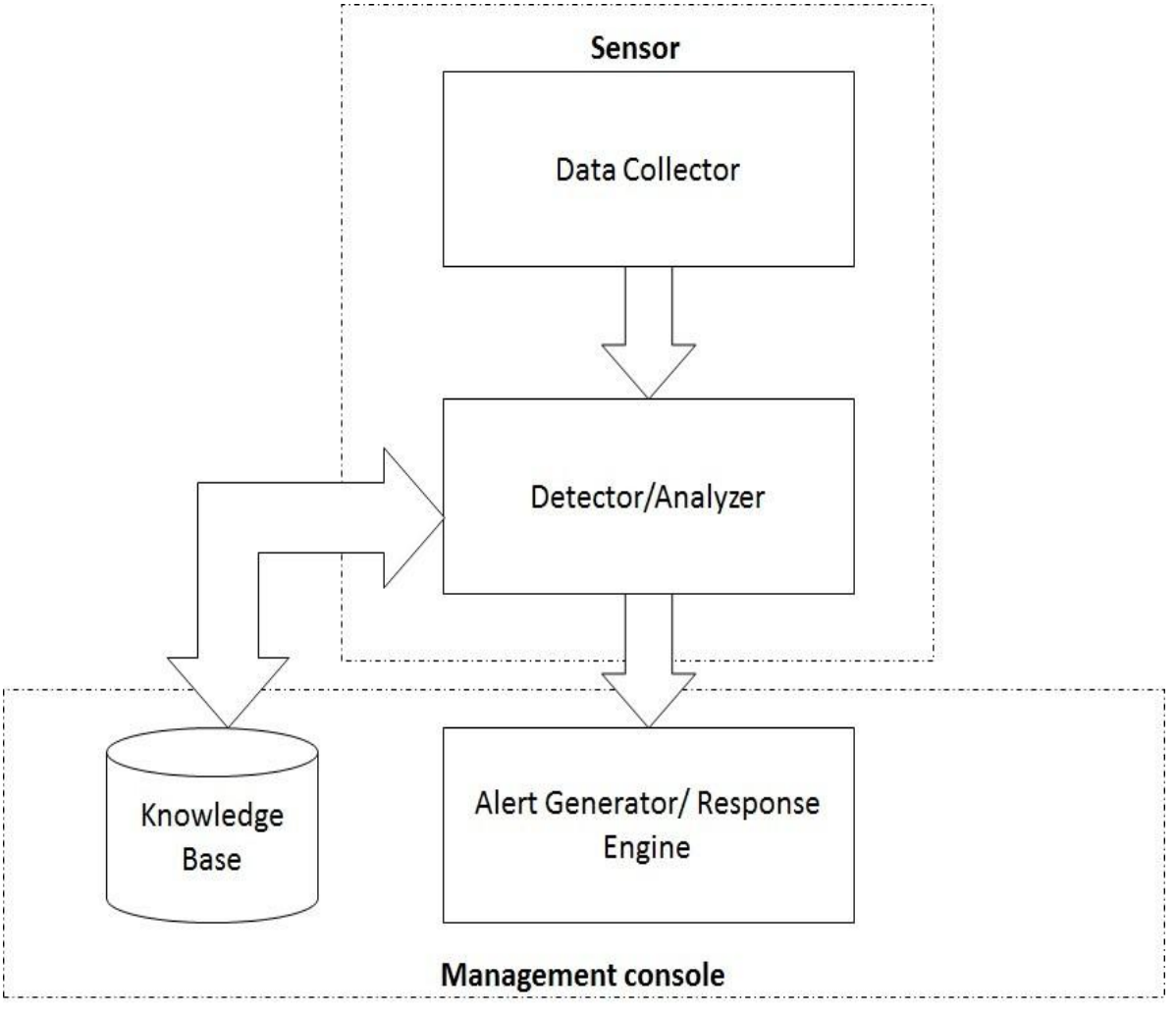

Figure 2: Components of Intrusion Detection System

Cybersecurity is the study of " the defence of cyberspace systems, data, and networks ". According to wikipedia, it is the "virtual space created by the interaction of users of software, services, and networks connected to the Internet". Installing systems for detecting intrusions allows for the achievement of one crucial aspect of system and network security. IDS scan networks or systems for illegal activity or infractions and send out notifications when anything suspect is found. IDS development went through various phases. These phases changed as people's reliance on technology and automation increased, and machine learning (ML) and deep learning (DL) techniques made significant strides. A class of neural networks known as deep learning (DL) use numerous layers to extract higher-level characteristics, enabling the modelling of complicated issues. IDS are systems designed to track and examine other systems and/or network traffic. The identification of anomalies, intrusions, or privacy violations is the aim of IDS. IDS come in two varieties: host intrusion detection systems (HIDS) and network intrusion detection systems (NIDS). Figure 3 shows the two types as they differ in their monitoring scope. NIDS keep an eye on the communication taking place within a network or among its sub-networks. They look at the traffic flow and internal and external communication.

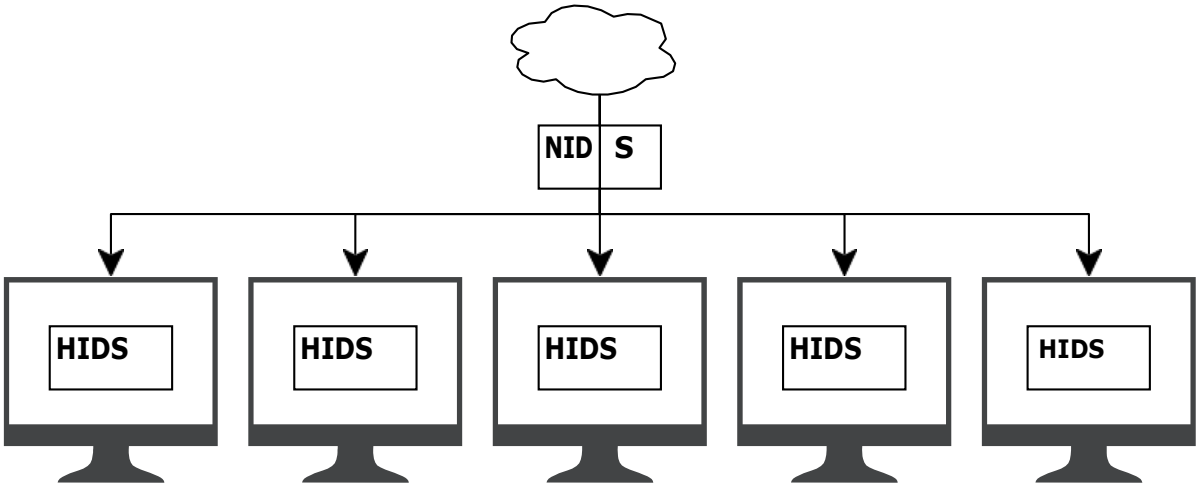

Figure 3: IDS Types

A traffic flow is defined by the packets used in communication between two network nodes. When the source and destination Internet Protocol (IP) addresses are provided, a network flow could be a 2-tuple. When the source and destination ports are used, a flow is referred to as a 4-tuple, while a 5-tuple flow additionally contains the protocol. Traffic flows can be either unidirectional or bidirectional. Both anomaly-based and signature-based IDS exist. In order to recognise known incursions and attacks, the "Misuse Detection" method of signature-based IDS uses prepared signatures. As a result, assaults can be detected by signature-based IDS by contrasting with recognised signatures. However, the database used to store the signatures limits their capacity to identify specific sorts of attacks, including zero-day (unknown), undetectable assaults.

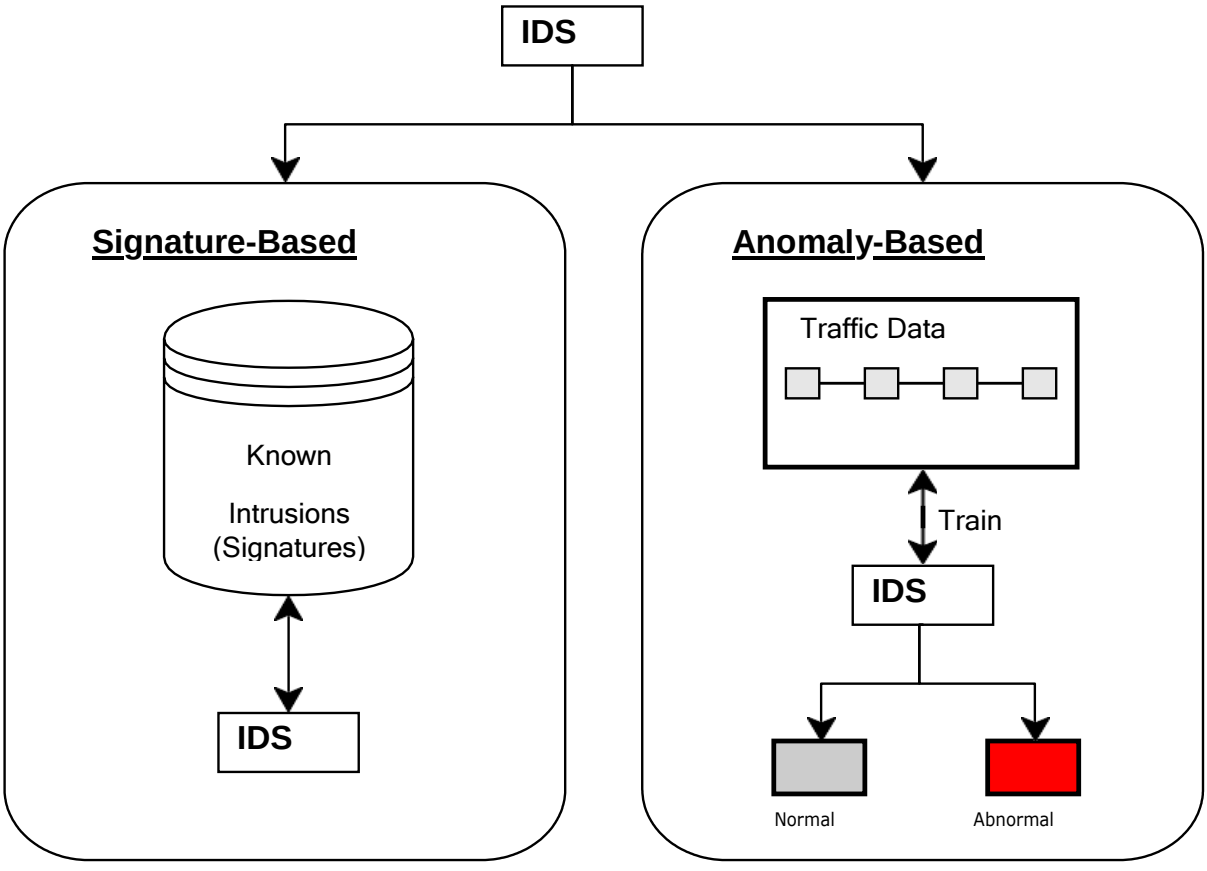

Figure 4 : Signature-based versus Anomaly-based IDS

The "Behaviour-based Detection" method, which is also known as anomaly-based IDS, relies on pattern recognition. This method requires training the system first. Due to their extensive training capabilities, artificial intelligence (AI) techniques, particularly ML and DL, are well suited for anomaly-based IDS. The advantage of anomaly-based IDS is their ability to distinguish between normal and abnormal traffic, thereby identifying both known and unknown assaults. Anomaly-based IDS is more effective against unidentified assaults than signature-based IDS. The False Positive Rate (FPR) is, however, frequently high. Specification-based IDS attempts to build a hybrid model that

can attempt to recognise both known and unexplained threats using a variety of AI techniques by combining the benefits of both anomaly-based and signature-based approaches. Figure 4 shows the contrast between anomaly-based and signature-based IDS. IDS that are anomaly-based and signature-based can function statefully or statelessly. While stateful IDS depends on network flows, stateless IDS depend on packets. Modern IDS are stateful because they take advantage of the "context" that flows offer. The distinction between IDS and Intrusion Prevention System (IPS), which can also perform corrective and preventive actions, should be made. IDS are responsible for detecting intrusions.

Through the detection and prevention of harmful activity, Network security is significantly impacted by systems for detecting intrusions. Due to the lack of data for model training and detection due to the dynamic and time-varying network environment, a substantial rate of false detection occurs when the network intrusion data are combined into a vast number of normal samples. A key component of stable services in extended network contexts like big data and the Network security's key technology, the network intrusion detection system, monitors packets for potentially hazardous activity taking place on the network. Big data has made it easier for people to live an information-based Internet lifestyle, but it has also created a number of serious network security issues that make it difficult to use networks on a regular basis. At the moment, systems for detecting intrusions are mostly used to identify aberrant network traffic. The actual process of the current traffic incursion detection systems needs to be changed, nonetheless, due to their numerous flaws and high resource occupation rate. While these network systems for detecting intrusions

and machine learning and deep learning have been explored simultaneously, performance in the real world or the class balance issue have not yet been resolved. The academic community has been concentrating on identifying network assaults that target information and communication systems for years. A complex issue, network intrusion detection poses a variety of difficulties. While novel attacks appear as a result of the expansion of connected devices and the advancement of communication technology, many current attacks go unnoticed. Systems for detecting network intrusions are crucial for safeguarding recent communication systems. This equipment was first hard-coded to recognise particular signatures, patterns, and rule infractions; however, algorithms for artificial intelligence and machine learning increasingly provide viable substitutes. However, a variety of out-of-date datasets and a wide range of different evaluation measures are employed in the literature to demonstrate algorithm effectiveness. In recent years, network dangers and hazards have been developing quickly. Networks are protected by a variety of technologies (firewalls, anti-virus software, anti-malware software, and spam filters). A robust and efficient network security solution for spotting unwanted and unusual network traffic flow is an systems for detecting intrusions. This new design has network control and managing capabilities. Despite this, the network's intruder flow makes it difficult to reap the benefits. In order to lessen the impact of invaders, the study issue of intrusion detection and prevention system (IDPS) has attracted attention. A targeted assault known as a distributed denial of service (DDoS) emerges when malicious traffic is flooded into a specific network device. Even with genuine network devices, these intruders can compromise the authenticated device and inject malicious traffic.

Several research works have been proposed in the literature for network traffic based intrusion detection such as [1], [2], [3], [4], [5], and many others. However, due to the inclusion of irrelevant features in their study, they report a significant false positivity rate. In this research, we have suggested a Statistical Analysis Based Intrusion Detection System by employing Machine Learning (ML) approach with the goal of ranking the traffic data and so eliminating the irrelevant features, which may limit the detection accuracy. A substantial amount of network traffic data that includes both malware data and normal data is gathered in order to identify the patterns of normal and malware data. Raw data is filtered to remove metadata and TCP flows are extracted from each captured file. Further, to rank the features, we applied the T test with two samples assuming unequal variances statistical analysis on nine different kinds of traffic features that comprise both malware and regular data, producing nine "t" values, one for each feature. We ranked the features based upon this obtained "t" value. Such a ranking of traffic features helps us in eliminating the irrelevant features while testing. Then Naive Bayes machine learning algorithm was applied to the testing data with 9 features by deleting each feature individually (removing from the last ranked feature), yielding nine distinct accuracy scores. After examining the accuracy value, we get to the conclusion that the three features with the lowest "t" value are removed in order to attain the highest accuracy, with accuracy of the data being increased as we removed these lower ranked features.

Chapter 2: Related Work

This section examines the related work of intrusion detection. Several works have been reported in the literature for intrusion detection on mobile platforms such as [24], [25], [26], [27], [28], [29], and [30]. Out of these, works like [24], [26], and [31] focused on capturing network traffic of Android samples and found the distinguishing features, i.e., the features that can efficiently detect malicious Android network traffic. Authors in these works applied machine learning techniques on network traffic features to detect malicious network activity in smartphones. However, our work is based on desktop-based intrusion detection, hence, in this section; we focus in detail on desktop-based intrusion detection.

The authors in [1] had conducted a sample test to determine the network's invasion behaviour. The simulation outcomes showed that the approach proposed in their study had higher detection accuracy, a higher true positive rate, and a lower false positive rate. LeNet-5 and DBN had detection accuracies of 8.82% and 0.51% respectively greater than that of the conventional models, respectively, according to the test results on the test set KDDTest + in their article.

Pontes et al. [2] proposed a novel technique they refer to as the Energy-based Flow Classifier (EFC). This statistical model is inferred from labelled benign examples using inverse statistics by an anomaly-based classifier. The

researchers demonstrated that EFC is more flexible to varied data distributions than conventional ML-based classifiers and can accurately conduct binary flow classification. Given the favourable outcomes on three distinct datasets, they thought that EFC is a potential technique for doing robust flow-based traffic categorization on datasets of CIDDS-001, CICIDS17, and CICDDoS19.

The weighted precision, recall, and F1 scores were used to compare FL with their [3] approach using a dataset collected from 20 widely distributed networks over the course of 60 days. It is shown that Segmented-FL performs better in all three categories of intrusion detection tasks using an analysis of Segmented-optimized FL's hyperparameters and three different evaluation methods, with validation weighted F1 scores of 0.964, 0.803, and 0.912 for Methods A, B, and C, respectively.

To stop TCP port scanning attacks, Bertoli et al [4] tested the AB-TRAP in both local (LAN) and international (internet) contexts. For the LAN study scenario, a decision tree with minimal CPU and RAM usage in kernel space yielded a f1-score of 0.96 and an area under the ROC curve of 0.99. A single-board computer with an average f1-score of 0.95, an average area under the ROC curve of 0.98, and an average overhead of 1.4 percent CPU and 3.6 percent RAM, the internet case uses eight machine learning methods.

Even in the presence of sampling, it can still offer a reliable assessment of NIDS. The authors in [5] discovered through sample studies that even at low

sampling rates like 1/10 and 1/100, malicious flows with smaller size (i.e., number of packets) are likely to go undetected. Next, they [5] looked at the effects of different sampling approaches on the NIDS detection rate and false alarm rate using the assessment method that was suggested. Three sample rates—1/10, 1/100, and 1/1000—four different sampling methodologies, and three classifiers—two tree-based and one based on deep learning—were used to calculate the detection rate and false alarm rate.

Kim et al [6] demonstrated that network incursion outside the area of the learnt data in the feature space can evade the ML-NIDS by examining the learning characteristics using representative features. Designing the active session to be classified early, before it leaves the training dataset of the ML-detection NIDS's range, can successfully stop this from happening.

The authors in [7] developed data-driven IDS by analysing the link load behaviours of the Road Side Unit (RSU) in the IoV against various assaults that result in irregular variations in traffic flows. To identify intrusions aimed at RSUs and extract link load features, a Convolutional neural network (CNN)-based deep learning architecture is utilised. Due to the convergence of the backpropagation method, the suggested approach includes a standard CNN and a basic error term.

He et al. [8] here proposed that real procedure of the current traffic intrusion detection systems has to be improved due to their numerous flaws and high

resource occupation rate. The classic traffic detection technique is improved in this study using deep learning technology and a clustering algorithm.

The data was first preprocessed, then 62 features were extracted, and then Zhou et al [9] suggested C4.5 divided algorithm was used to detect traffic. The CSE-CIC-IDS2018 public data collection was used in this experiment for verification. The results of the experiments show how effective the technique described in this article in identifying various cyberattacks.

Ageyev et al [10] study presents a classification of datasets into categories such network traffic-based dataset, internet traffic-based dataset, virtual private network-based dataset, and IoT traffic-based dataset, along with a brief explanation of some of the most well-known datasets. System for detecting intrusions relies heavily on dataset.

In this research [11], as part of early development of abnormalities in the smart home network are found utilising the Extreme Learning Machine and Artificial Immune System (AIS-ELM) intrusion detection system (IDS). The input parameters are evaluated by ELM for improved convergence in detecting anomalous behaviour after being optimised by AIS using the clonal approach.

Hashemi et al [12] provided Reconstruction from Partial Observation (RePO), a novel method for developing a network intrusion detection system (NIDS) that

uses denoising autoencoders to identify various types of network attacks with less false alarms and improved resistance against adversarial example attacks. In comparison to other recently proposed anomaly detectors, their [12] analysis of Denoising autoencoders can improve the identification of fraudulent traffic by up to 29 percent in a typical environment and by up to 45 percent in an adversarial context, according to a dataset featuring a variety of network threats.

To assess the model's efficacy, they [13] use the CSE-CIC-IDS2018 dataset. To compare the performance of various models, the authors in this [13] select evaluation indicators and compare our model to the other five methods. According to experimental findings, The ACNNBN model's performance is unquestionably better than that of the other five models in the evaluation index and can greatly increase detection accuracy.

Rose et al [14] here presented that, raw traffic is sent to the machine learning classifier for analysis and the detection of potential attacks. On the Cyber-Trust testbed, the proposed methodology's performance is evaluated using legitimate and harmful internet activity. The testing results show the potential of the suggested anomaly detection system, with a 98.35 percent overall accuracy and a 0.98 percent false-positive alert rate.

Raju et al [15] used three deep learning models, and they [15] used a semi-balanced version of the CIC-IDS2017 and CSE-CIC-IDS2018 dataset to assess

the outcomes. The Fully Connected Network, Seq2Seq LSTM, and Autoencoder models' performance was evaluated using standard metrics. They [15] introduce two new assessment metrics—Matthews Correlation Coefficient and Cohen's Kappa Coefficient—to assess how well the employed schemes performed because the well-known conventional metrics were unable to provide any conclusive findings.

The authors here [16] by evaluating giving foundation knowledge on either deep learning or machine learning algorithms for systems for detecting intrusion from the literature, the major goal of this [16] study is to survey in-depth learning and machine learning approaches for intrusion detection. On the DARPA dataset, the paper also evaluates the efficiency of various machine learning categorization techniques.

The authors here [17] proposed the study that in all those cases when it is difficult to understand the process of interest rationally, machine learning [ML] technologies are being applied more and more. Numerous techniques based on ML methodologies are now being developed. In networked systems, intrusion detection is a problem when it is crucial to get an answer a classification algorithm can determine whether abnormalities are affecting the network traffic, even while it is not necessary to interpret the measurements collected from a process.

The authors in [18] offered a novel network intrusion detection model based on convolution neural networks (ACNNBN). Convolution Block Attention Module is a new feature introduced by the ACNNBN to enable cross-channel fusion with convolution modules. This feature enables the ACNNBN to swiftly identify information that is more important for the job at hand. Additionally, the model can discover Prevent vanishing gradients, hasten model convergence, and certain key shallow layer properties that the deeper layer of CNN might miss.

The implementation of an anomaly-based network intrusion detection system employing stacking and boosting ensemble methods is the main goal of this [19] work. The same dataset, known as the Both approaches are implemented using the NSL-Knowledge Discovery Dataset (NSL-KDD), a recognised benchmark in the field of research of intrusion detection strategies.

Qadeer et al [20] here by using an existing IDS, SNORT, an efficient multicore approach for network monitoring has been put forth. The suggested architecture has a total of 16 cores to effectively process Ethernet traffic. Additionally, the suggested architecture uses PF ring for effective packet capturing and IP hash load balancing for network load balancing. With "N" times, where N is the number of active CPU cores, network monitoring can be enhanced based on the aforementioned variables.

Baig et al [21] suggested a method for distinguishing between regular and abnormal network traffic based on GMDH. On the KDD 99 dataset, two variations of the technique—monolithic and ensemble-based—were put to the test. The dataset underwent pre-processing, and all features were ranked using the Information Gain, Gain Ratio, and GMDH by itself feature ranking approaches. Results showed that, when compared to previous intelligent classification algorithms for network intrusion detection, the proposed intrusion detection methodology yields high attack detection rates, close to 98 percent.

Finding the most relevant and practical characteristics that can be used as key features in a brand-new IDS dataset was the aim of this [22] study. To achieve the objective, a method for constructing an optimal IDS ensemble is created. The following six feature selection techniques are utilized and contrasted: Chi-Square, Relief-F (R-F), Symmetrical Uncertainty (SU), One-R (OR), and Gain Ratio (GR) are all metrics for information gain (CS). Collections of selected characteristics are produced by the feature selection methods. The best number of features from each feature selection methodology's feature ranking stage would be used to categorize assaults using the following four traditional classification methods: Decision Tree: J48, Naive Bayesian (NB), SOM, and Bayesian Network (BN). The best features from each feature selection methodology and the best features from each classification methodology are merged to create ensemble IDSs. Finally, the Hold-up, K-fold, F-Measure, and statistical validation procedures are used to analyze the ensemble IDSs. The efficient ensemble IDSs using (SU and BN), (CS and BN), (CS and SOM), (IG and NB), and (OR and BN) with respective ten, four, and seven better-selected

features achieve 81.0316 percent, 85.2593 percent, and 80.8625 percent of accuracy, according to experimental results using Weka tools on the ITD-UTM dataset.

NSL-KDD is used to assess the machine learning techniques for intrusion detection. However, not all traits are beneficial for performance. In order to increase speed and accuracy, a specific collection of features might be reduced or chosen. Recursive Feature Elimination is therefore used to choose the features (RFE). Their [23] thorough investigation on the Intrusion Detection System (IDS), which employs the machine learning techniques Random Forest and Support Vector Machine (SVM). They [23] have shown how the performance of the model before and after Random Forest and SVM feature selection may be compared. Additionally, they [23] have provided the confusion matrices.ach sign in the dataset, which is not acceptable for real-time SLR systems.

The authors in [32] proposed an anomaly detection method based on features importance. They applied Principal Component Analysis, and Clustering based approaches on the BETH dataset to efficiently detect anomalies. Raman et al. [33] applied several feature selection techniques such as Pearson Correlation, Information Gain, ExtraTreeClassifier, and Chi- Square tests to rank the features and further applied machine learning techniques for intrusion detection. Yash et al. [34] applied ANOVA and Chi-Square Test to rank the traffic features, however, the authors did not propose a detection model. In this

work, we have applied Student's T-Test for feature ranking and we have also proposed an intrusion detection model. The details of the proposed work are presented in next section.

Haystack IDS [39] uses pattern matching based on attack profiles to cut down on lengthy audit trails. Up until the year 1990, IDS work was solely focused on a single host machine. Extending intrusion detection is an idea from the host to the local network and from the local network to arbitrarily larger networks was first introduced by researchers in [40], who also introduced the distributed intrusion detection system. Both a networked system with numerous hosts and a single system could be monitored by the distributed IDS [40].

Network anomaly detector and intrusion reporter [41], the first anomaly IDS based on statistics-based expert system, was developed by the researchers, under the impact of the IDS paradigm presented by denning [42]. In [43], Vern Paxson announced the Bro IDS, which supports its own ruleset language for analysing traffic via the libpcap packet capture library.

Amoroso [44] used libpcap to create the Network Flight Recorder (NFR) tool in order to overcome the security and network administration challenges. Initially created as a packet sniffer tool in [45], APE later went by the name snort. With three lakh active users worldwide, Snort has now surpassed other signature-based IDS systems in terms of usage. The intrusion detection systems in use in the late 1990s were unable to identify new types of attacks. This provided the

framework for anomaly-based intrusion detection systems, which learn the system's typical behaviour and identify any change as an assault.

Anup K. Ghosh et al. presented a method to learn the typical intrusion detection profile in [46] with the aim of observing and detecting unusual traffic. When evaluating an intrusion detection system offline against the DARPA'99 dataset, which served as a benchmark dataset, Lippmann [47] demonstrates an approach.

Mahoney suggested a method in [48] based on the packet bytes and packet header anomaly detector (PHAD) for anomaly identification from network traffic.

An alert correlation cooperation module was put up by Cuppens [49] to correlate the alerts into a single scenario. In order to evaluate the total network attack scenario, an intelligent intrusion detection system that combines the inference from abuse and anomaly-based intrusion detection methods with a fuzzy model.

Based on the Dempster-Shafer theory, Yu and Frincke [50] created a framework to combine the alerts and accurately identify intrusions. Their research demonstrates the alert confidence metric evaluation technique, which was based on minimal mean square error and maximum entropy.

An anomaly detection technique was put into practise by Aickelin [51] using the Dempster-Shafer fusion rule for fusing features. Their method was able to handle scenarios where some anomaly detecting features were lacking or ineffective.

Chapter 3: Proposed Methodolgy

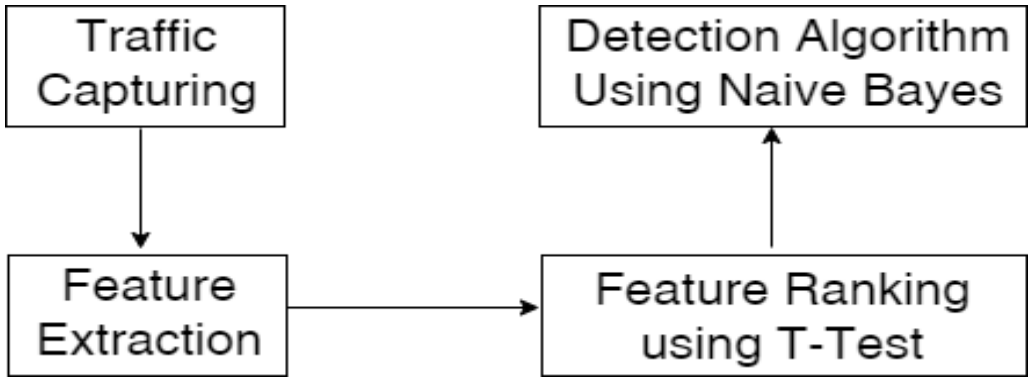

Figure 5 summarizes the proposed methodology of this work. The work is divided in sub phases and we have discussed each phase in detail in the upcoming subsections.

3.1 TRAFFIC CAPTURING:

The well-known network packet capture programme, Wireshark, is one of the best resources for IT workers. You may record network packets with Wireshark and view them in-depth. These packets can be utilised for offline or real-time analysis after being broken down. You can carefully watch your network traffic with the help of this application, filter it, and go further to find any issues. Furthermore, this technology supports network analysis, which in turn promotes network security.

A tool called Wireshark, for example, is a network protocol analyzer that keeps track of the packets sent over connections like the one connecting your

computer to the internet or your home office. A discrete piece of data in a standard Ethernet network is known as a packet.

Wireshark is the name of the most widely used packet sniffer worldwide. Wireshark accomplishes three tasks, similar to other packet sniffers:

1. **Packet Capture:** By continuously monitoring a network connection and collecting complete streams of data, Wireshark may be able to capture several hundred thousand packages at once.

2. **Filtering**: Filters in Wireshark can be used to slice and dice all of this arbitrary live data. You can only see the data you need if you apply a filter.

3. **Visualization**: Like any good packet sniffer should, Wireshark allows you access to a network packet's internal working. Using this, you can also view whole chats and network feeds.

Using the Wireshark software that is installed on our PC, we were able to collect the typical normal network traffic. Additionally, the Canadian Institute for Cyber Security provided us with the intrusion traffic. We recovered TCP flows from each of the pcap files containing the traffic that was recorded. Our research study employed about 35,000 normal traffic flows and 30,000

malware traffic flows. Nine traffic features were taken from each traffic flow, and their specifics are provided in the following section.

3.2 FEATURE EXTRACTION:

The method of feature extraction keeps the information from the original data collection while transforming raw data into manageable numerical features. It yields better results in comparison to performing machine learning on the raw data directly. By linearly integrating the preexisting features, we can produce new features with the aid of the feature extraction technique. The values of the new set of features will differ from the values of the original characteristics. The fundamental objective is to use fewer features to collect the same information. The main difference between the two is that feature extraction adds new features whereas feature selection just chooses a subset of the original feature set. Feature selection is a method for reducing the input variable for the model by choosing only relevant data in order to decrease overfitting in the model. Nine traffic features that we extracted are described in the Table 1 below.

Table 1: List of traffic features used along with their notations

S. No.	Feature	Notation
1	Average packet size	F1
2	Average packet size sent	F2
3	Average packet size received	F3
4	Average time interval between packets sent	F4
5	Average time interval between packets received	F5
6	Ratio of incoming to outgoing packets	F6
7	Ratio of incoming to outgoing bytes	F7
8	Average flow duration	F8
9	Average number of destinations per connections	F9

1. **Average packet size:** This feature is determined by the packet's size. By gathering every packet, the average size of the packets is determined. This feature's notation is F1.
2. **Average packet size sent :** This function is based on the packet number that was sent. By accumulating all the packets, the average number of packets sent is determined. This feature's notation is F2.
3. **Average packet size received :** This feature is based on the quantity of packets that were received via packet capture. By assembling all the packets, the average number of packets received is calculated. This feature's notation is F3.
4. **Average time interval between packets sent** : This feature is based on how long it takes for packets to arrive one after the other when dispatched. All of the packets are being sent, which takes an average amount of time. This feature's notation is F4.
5. **Average time interval between packets received :** This feature is dependent on how long it takes to get each packet one after the other. The total number of packets received determines the average time required to process each packet after receipt. This feature's notation is F5.
6. **Ratio of incoming to outgoing packets :** The basis for this feature is the ratio of total packets received to total packets sent. The ratio of packets sent to packets received calculated by sending and receiving every packet. This characteristic is denoted as F6.
7. **Ratio of incoming to outgoing bytes :** This feature is based on the weight that the packets are carrying. The ratio of total bytes received to total bytes transmitted, calculated by sending and receiving each packet individually. This feature's notation is F7.
8. **Average flow duration :** The speed at which packets are sent and received determines the functionality of this feature. the

average flow of all packets transmitted and received when they are all sent and received at various rates. This feature's notation is F8.

9. **Average number of destinations per connections :** Based on the packets that arrived at the destination in accordance with their connections, this functionality was developed. The average number of packets that are delivered to each connection's destination are based on the packets that are received at the connection's destination source within a specific time period. This characteristic is denoted as F9.

3.3 FEATURE RANKING USING T-TEST:

The task of measuring the effects of specific input features (variables) on how well a supervised learning model performs is known as feature importance ranking (FIR) in machine learning. FIR has emerged as one of the most effective methods in explainable/interpretable AI for comprehending decision-making by a learning system and identifying crucial elements in a particular field, such as in medicine for determining which genes are probably the primary causes of a cancer. Feature selection is widely used to increase the generalisation of a learning system and to handle the well-known curse of dimensionality difficultyDue to the presence of correlated/dependent and irrelevant features to objectives in high-dimensional actual data, a subset of optimal features is chosen in accordance with the pre-defined criteria to optimise the performance of a learning system. Both population-level and instance-level feature selection are possible; population-level approaches would

identify an ideal feature subset for all examples within a population, but instance-level methods are more likely to identify a subset of salient features unique to a single instance. By ranking the significance of those features in an ideal subset, FIR is always closely related to feature selection in practise and can also be used as a stand-in for feature selection.

This application does a two-sample student's t-test on data sets from two distinct populations with different variances. This test can be two-tailed or one-tailed depending on whether we're determining whether the two population means differ or if one is higher than the other. When A two-sample T-test with unequal variance can be employed if the samples are normally distributed, the standard deviations of the two populations are unknown and presumed to be unequal, and the sample size is large enough (over 30). Even though The variance is uneven and the standard deviations vary The confidence interval for the difference between two means is provided by the t test. The t test with unequal variance is calculated. In both t tests, a “t” value and confidence interval are reported. The 10 different p values are obtained after the t test on the 10 data set. The data set will be categorised in decreasing order after analysis to determine the p value. The various data set is designated as F1 to F9. They will be sorted according to the “t” value obtained after applying the t test, in descending order.

S. No.	X	Y
1	2	1
2	0	9
3	6	5
4	5	0
5	9	0

Suppose X and Y, as summarized in the table shown above represent the values of any traffic feature "F" for normal and intrusion traffic respectively, then the

Formula used for the T-test is given below:

$$T = \frac{(\sum D)/_N}{\sqrt{\frac{\sum D^2 - (\frac{((\sum D)^2)}{N})}{(N-1)(N)}}}$$

Where:

1. The "ΣD" is the sum of X-Y.
2. ΣD2: Sum of the squared differences.
3. (ΣD)2: Sum of the differences , squared.

3.4 DETECTION ALGORITHM USING NAÏVE BAYES CLASSIFIER:

- Based on the Bayes theorem, the Nave Bayes algorithm is a supervised learning technique for classification problems.
- It primarily uses a huge training set for text categorization..
- The Naive Bayes Classifier is one of the simplest and most effective classification algorithms on the market right now. It facilitates the creation of efficient machine learning models that can produce reliable predictions.
- Because it uses a probabilistic classifier, it bases its predictions on the likelihood that a given event will take place.

- Some applications for Naive Bayes algorithms include spam filtration, sentiment analysis, and article classification.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- **Naïve**: It is referred regarded as naïve since it assumes that the existence of one trait is unconnected to the prevalence of other features. For instance, if a red, spherical, sweet fruit is recognised as an apple based on its colour, shape, and sweetness. As a result, each quality, without depending on the others, aids in identifying it as an apple.
- **Bayes**: It is named as Bayes because it relies on the Bayes' Theorem premise.

Bayes' Theorem:

- Bayes' theorem, also referred to as Bayes' rule or Bayes' law, is a technique for determining how likely a hypothesis is given some prior knowledge. The conditional probability determines this.
- The formula for Bayes' theorem is given as:

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

The NAÏVE BAYES classification algorithm is a probabilistic classifier. It is based on probability models that make significant independence assumptions. Often, the independence assumptions have no impact on reality. Consequently, they are seen as being naive. NAÏVE BAYES is a machine learning model that is used for vast amounts of data; it is the suggested strategy even when using data that has millions of records. It does NLP tasks like sentimental analysis with excellent results. On the 9 features data set that we recovered from the raw network traffic, we applied the NAÏVE BAYES algorithm. The nine features are subjected to the NAÏVE BAYES machine learning method, and each feature will be eliminated one at a time based on its ranking. We start removing the feature from the last, i.e., lowest ranked feature will be eliminated and we test for detection accuracy. We repeat this procedure till we get the highest detection accuracy. We report the outcomes of the suggested strategy are presented in the following section.

Chapter 4: Results and Discussions

In this section, we emphasize the outcomes of the proposed approach that was used to implement the statistical analysisbased intrusion detection system. We applied the T-test: two sample assuming unequal variances statistical analysis to rank the features. To identify the pattern of the malware data, a sizable volume of network traffic data that contains both malware data and regular data is collected. The t-test is run on 9 different types of data sets. Following the application of the t-test to nine distinct data sets that comprise both malware and normal data, nine t-values were created. Table III shows the feature ranking based on the decreasing "t" value. The higher the "t" value, the is better the feature with distinguishing capability. As can be seen from table III, the feature F3, i.e., Average packet size received is the most distinguishing feature between normal traffic and intrusion traffic. Next to F3 is the feature F2, i.e., the Average packet size sent is second in the ranking. The worst or most irrelevant feature turns out to be the Ratio of Incoming to Outgoing Bytes with the least "t' value.

Table 2: Feature Ranking based on T-Value of features

S.No.	Features	T-Value
1	F3	115.2452
2	F2	112.7775
3	F5	93.47604
4	F4	84.80577
5	F9	76.41087
6	F8	58.26489
7	F1	49.03203
8	F6	10.7755
9	F7	10.23431

Table 3: Detection at Different Iterations

Feature Set	Naive Bayes Accuracy
All 9 features	**0.7613846153846154**
Top 8 features	**0.7896410256410257**
Top 7 features	**0.9569230769230769**
Top 6 features	**0.9456410256410256**
Top 5 features	**0.9406666666666667**
Top 4 features	**0.9428717948717948**
Top 3 features	**0.9424102564102564**
Top 2 features	**0.9320512820512821**
Top 1 features	**0.9023589743589744**

Further, we applied our proposed approach using the NAÏVE BAYES machine learning method to 9 data sets, it produces 9 different accuracy values by removing 1 feature at a time. We removed the last ranked feature, checked the accuracy, and if the accuracy increased, we permanently remove that feature. Then we proceed toward the next lowerranked feature and the process continues. Table IV summarizes these detection results for different iterations. As can be seen from Table IV, the top 7 features give us the highest detection accuracy of around 95.69%. If we further remove the features, the accuracy decreases. Hence, we can say that we get the best accuracy of 95.69%. We can argue that feature ranking helps us in improving detection accuracy. As can be seen from Table IV, if do not apply feature ranking and use all 9 features for detection, we get an accuracy of around 76%. And it increases

significantly to 95.69% on removing the lower two ranked features. Hence, feature ranking using the T-test helps us in improving detection accuracy. The accuracy number will be produced after the NAÏVE BAYES theorem has been applied to the data sets. After removing each feature individually, the various accuracy values are shown in table IV. So, as we can see in table IV when the NAÏVE BAYES algorithm is applied to all 9 characteristics, the accuracy value is lowest. The accuracy value decreases after 7 features, so based on this outcome, we can conclude that the maximum accuracy is provided when all 7 features are present.

A. Comparison with other works

In this subsection, we compare our proposed approach with other similar works in the field of network traffic based intrusion detection. Table V summarizes the comparison of our work with other similar works. As can be seen from the Table V, our work outperforms several other similar works proposed to detect intrusions based upon network traffic.

TABLE IV
COMPARISON OF PROPOSED WORK WITH OTHER SIMILAR WORKS

Related Work	Detection Accuracy
Hsieh et al. [35]	79%
Chen et al. [36]	93.56%
Shah et al. [37]	95%
Chindove et al. [38]	90%
PROPOSED WORK	**95.69%**

Chapter 5: Conclusion and Future Work

In this work, we proposed an intrusion detection system based on the two sample assumption of unequal variances T test, a statistical analysis of ranking approach. In order to determine the pattern of the malware data, a sizable amount of network traffic data that consists of both intrusion and normal data was obtained. The T-test was used to score nine different traffic features that were retrieved from both intrusion and normal traffic files. Based upon the "t" value obtained from the T-test, features were ranked. The Naive Bayes machine learning algorithm was next used on the nine features, eliminating the ones with the lowest t-values one at a time to provide nine different accuracy values. After looking at the accuracy value, we got to the conclusion that in order to achieve the best accuracy, the two features with the lowest value should be removed. Instead of using all nine features, our work's accuracy rate of 95.69% was attained using the top 7 features. In our future work, we will aim to increase the number of traffic features for analysis and detection.

BIBLIOGRAPHY

[1] H. Yang and F. Wang, "Wireless Network Intrusion Detection Based on Improved Convolutional Neural Network," in IEEE Access, vol. 7, pp. 64366-64374, 2019

[2] C. F. T. Pontes, M. M. C. de Souza, J. J. C. Gondim, M. Bishop and M. A. Marotta, "A New Method for Flow-Based Network Intrusion Detection Using the Inverse Potts Model," in IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1125-1136, June 2021.

[3] Y. Sun, H. Esaki and H. Ochiai, "Adaptive Intrusion Detection in the Networking of Large-Scale LANs With Segmented Federated Learning," in IEEE Open Journal of the Communications Society, vol. 2, pp. 102-112, 2021

[4] G. De Carvalho Bertoli et al., "An End-to-End Framework for Machine Learning-Based Network Intrusion Detection System," in IEEE Access, vol. 9, pp. 106790-106805, 2021.

[5] J. Alikhanov, R. Jang, M. Abuhamad, D. Mohaisen, D. Nyang and Y. Noh, "Investigating the Effect of Traffic Sampling on Machine Learning-Based Network Intrusion Detection Approaches," in IEEE Access, vol. 10, pp. 5801-5823, 2022.

[6] T. Kim and W. Pak, "Robust Network Intrusion Detection System Based on Machine-Learning With Early Classification," in IEEE Access, vol. 10, pp. 10754-10767, 2022.

[7] L. Nie, Z. Ning, X. Wang, X. Hu, J. Cheng and Y. Li, "Data-Driven Intrusion Detection for Intelligent Internet of Vehicles: A Deep Convolutional Neural Network-Based Method," in IEEE Transactions on Network Science and Engineering, vol. 7, no. 4, pp. 2219-2230, 1 Oct.-Dec. 2020.

[8] Q. He, "Research on Network Traffic Anomaly Detection Based on Deep Learning," 2021 International Conference on Networking, Communications and Information Technology (NetCIT), 2021, pp. 50-53.

[9] J. Zhou, X. Jiang, C. Liu, J. Zhang, L. Liao and J. Lu, "Multi-Traffic Features Network Intrusion Detection Algorithm Based on C4.5," 2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2021, pp. 548-552.

[10] D. Ageyev, T. Radivilova, O. Bondarenko and O. Mohammed, "Data Sets Selection for Distributed Infocommunication Networks Traffic Abnormality Detection," 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 635-638

[11] E. D. Alalade, "Intrusion Detection System in Smart Home Network Using Artificial Immune System and Extreme Learning Machine Hybrid Approach," 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), 2020, pp. 1-2.

[12] M. J. Hashemi and E. Keller, "Enhancing Robustness Against Adversarial Examples in Network Intrusion Detection Systems," 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2020, pp. 37-43.

[13] L. Li, J. Mu, H. He and C. Liu, "An Attention-Based CNN with Batch Normalization Model for Network Intrusion Detection," 2021 China Automation Congress (CAC), 2021, pp. 3531-3536.

[14] J. R. Rose, M. Swann, G. Bendiab, S. Shiaeles and N. Kolokotronis, "Intrusion Detection using Network Traffic Profiling and Machine Learning for IoT," 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), 2021, pp. 409-415.

[15] D. Raju, S. Sawai, S. Gavel and A. S. Raghuvanshi, "Development of Anomaly-Based Intrusion Detection Scheme Using Deep Learning in Data Network," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021, pp.

[16] J. A. Abraham and V. R. Bindu, "Intrusion Detection and Prevention in Networks Using Machine Learning and Deep Learning Approaches: A Review," 2021 International Conference on Advancements in Electrical,

Electronics, Communication, Computing and Automation (ICAECA), 2021, pp. 1-4.

[17] S. Sridevi, R. Prabha, K. N. Reddy, K. M. Monica, G. A. Senthil and M. Razmah, "Network Intrusion Detection System using Supervised Learning based Voting Classifier," 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT), 2022, pp. 01-06.

[18] L. Li, J. Mu, H. He and C. Liu, "An Attention-Based CNN with Batch Normalization Model for Network Intrusion Detection," 2021 China Automation Congress (CAC), 2021, pp. 3531-3536.

[19] V. Sidharth and C. R. Kavitha, "Network Intrusion Detection System Using Stacking and Boosting Ensemble Methods," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), 2021, pp. 357-363.

[20] H. Qadeer, A. Talat, K. N. Qureshi, F. Bashir and N. Ul Islam, "Towards an Efficient Intrusion Detection System for High Speed Networks," 2020 17th International Bhurban Conference on Applied Sciences and Technology (IBCAST), 2020, pp. 428-433.

[21] Zubair A. Baig, Sadiq M. Sait, AbdulRahman Shaheen, “GMDH-based networks for intelligent intrusion detection”, Engineering Applications of Artificial Intelligence,
Volume 26, Issue 7,2013, Pages 1731-1740, ISSN 0952-1976.

[22] D. Stiawan et al., "An Approach for Optimizing Ensemble Intrusion Detection Systems," in IEEE Access, vol. 9, pp. 6930-6947, 2021.

[23] R. Patgiri, U. Varshney, T. Akutota and R. Kunde, "An Investigation on Intrusion Detection System Using Machine Learning," 2018 IEEE Symposium Series on Computational Intelligence (SSCI), 2018, pp. 1684-1691.

[24] A. Arora, S. Garg and S. K. Peddoju, "Malware Detection Using Network Traffic Analysis in Android Based Mobile Devices," Eighth International Conference on Next Generation Mobile Apps, Services and Technologies, pp. 66-71, 2014.

[25] A. Arora, S. K. Peddoju and M. Conti, "PermPair: Android Malware Detection Using Permission Pairs," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 1968-1982, 2020.

[26] A. Arora, and S.K. Peddoju, “Minimizing Network Traffic Features for Android Mobile Malware Detection”, in Proceedings of the 18th International Conference on Distributed Computing and Networking (ICDCN), article 32, pp. 1-10, 2017.

[27] A. Arora and S. K. Peddoju, "NTPDroid: A Hybrid Android Malware Detector Using Network Traffic and System Permissions," 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 808-813, 2018.

[28] A. Arora, S.K. Peddoju, V. Chouhan, and A. Chaudhary, “Hyrbid Android Malware Detection by Combining Supervised and Unsupervised Learning”, in Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, 2018.

[29] K. Khariwal, J. Singh and A. Arora, "IPDroid: Android Malware Detection using Intents and Permissions," Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), 2020, pp. 197-202, 2020.

[30] M. Upadhayay, A. Sharma, G. Garg and A. Arora, "RPNDroid: Android Malware Detection using Ranked Permissions and Network Traffic," Fifth World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), pp. 19-24, 2021.

[31] A. Shabtai, L. Tenenboim-Chekina, D. Mimran, L. Rokach, B. Shapira, and Y. Elovici, Mobile malware detection through analysis of deviations in application network behavior. Computers and Security, 43, pp. 1 – 18, 2014.

[32] N. Sushmakar, N. Oberoi, S. Gupta and A. Arora, ”An Unsupervised Based Enhanced Anomaly Detection Model Using Features Importance,” 2022 2nd International Conference on Intelligent Technologies (CONIT), 2022, pp. 1-7, doi: 10.1109/CONIT55038.2022.9848297.

[33] Raman, S. K. Jha and A. Arora, ”An Enhanced Intrusion Detection System Using Combinational Feature Ranking and Machine Learning Algorithms,” 2022

2nd International Conference on Intelligent Technologies (CONIT), 2022, pp. 1-8, doi: 10.1109/CONIT55038.2022.9847815.

[34] Y. Sharma, S. Sharma and A. Arora, ”Feature Ranking using Statistical Techniques for Computer Networks Intrusion Detection,” 2022 7th International Conference on Communication and Electronics Systems (ICCES), 2022, pp. 761-765, doi: 10.1109/ICCES54183.2022.9835831.

[35] C. -F. Hsieh and C. -M. Su, ”MLNN: A Novel Network Intrusion Detection Based on Multilayer Neural Network,” 2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), 2021, pp. 43-48, doi: 10.1109/TAAI54685.2021.00017.

[36] J. Chen, S. Yin, S. Cai, L. Zhao and S. Wang, ”L-KPCA: an efficient feature extraction method for network intrusion detection,” 2021 17th International Conference on Mobility, Sensing and Networking (MSN), 2021, pp. 683-684, doi: 10.1109/MSN53354.2021.00104.

[37] A. Shah, S. Clachar, M. Minimair and D. Cook, ”Building Multiclass Classification Baselines for Anomaly-based Network Intrusion Detection Systems,” 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), 2020, pp. 759-760, doi: 10.1109/DSAA49011.2020.00102.

[38] H. Chindove and D. Brown, ”Adaptive Machine Learning Based Network Intrusion Detection”, in Proceedings of the International Conference on Artificial Intelligence and its Applications (icARTi ’21). Association for Computing Machinery, New York, NY, USA, Article 15, 1–6. https://doi.org/10.1145/3487923.3487938, 2021.

[39] S. E. Smaha, “Haystack: An intrusion detection system,” in Aerospace Computer Security Applications Conference, 1988., Fourth. IEEE, 1988, pp. 37–44.

[40] S. R. Snapp, J. Brentano, G. V. Dias, T. L. Goan, L. T. Heberlein, C.-L. Ho, K. N. Levitt, B. Mukherjee, S. E. Smaha, T. Grance et al., “Dids (distributed intrusion detection system)-motivation, architecture, and an early prototype,” in Proceedings of the 14th national computer security conference, vol. 1. Washington, DC, 1991, pp. 167–176.

[41] J. Hochberg, K. Jackson, C. Stallings, J. McClary, D. DuBois, and J. Ford, "Nadir: An automated system for detecting network intrusion and misuse," Computers & Security, vol. 12, no. 3, pp. 235–248, 1993.

[42] D. E. Denning, "An intrusion-detection model," IEEE Transactions on software engineering, no. 2, pp. 222–232, 1987.

[43] V. Paxson, "Bro: a system for detecting network intruders in real-time," Computer networks, vol. 31, no. 23, pp. 2435–2463, 1999.

[44] E. Amoroso, "Intrusion detection: an introduction to internet surveillance, correla- tion, trace back, traps, and response," Intrusion. Net Book, 1999.

[45] S. Northcutt, Snort: IDS and IPS toolkit. Syngress Press, 2007.

[46] A. K. Ghosh, A. Schwartzbard, and M. Schatz, "Learning program behavior profiles for intrusion detection." in Workshop on Intrusion Detection and Network Moni- toring, vol. 51462, 1999, pp. 1–13.

[47] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham et al., "Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation," in DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings, vol. 2. IEEE, 2000, pp. 12–26.

[48] M. V. Mahoney and P. K. Chan, "Phad: Packet header anomaly detection for identifying hostile network traffic," Tech. Rep., 2001.

[49] F. Cuppens and A. Miege, "Alert correlation in a cooperative intrusion detection framework," in Security and privacy, 2002. proceedings. 2002 ieee symposium on. IEEE, 2002, pp. 202–215.

[50] D. Yu and D. Frincke, "Alert confidence fusion in intrusion detection systems with extended dempster-shafer theory," in Proceedings of the 43rd annual Southeast re- gional conference-Volume 2. ACM, 2005, pp. 142–147.

[51] Q. Chen and U. Aickelin, "Anomaly detection using the dempster-shafer method," in conference on data mining, DMIN'06. Citeseer, 2006.

AMITY UNIVERSITY

AMITY SCHOOL OF
ENGINEERING & TECHNOLOGY

Technical Co-sponsor
IEEE UP SECTION (INDIA)

Sponsors
DRDO
ICAR

G20 2023 INDIA

Ministry of Electronics and Information Technology
Government of India

13th International Conference on Cloud Computing, Data Science and Engineering

CONFLUENCE 2023

CERTIFICATE OF PARTICIPATION

This is to certify that Dr. / Mr. / Ms. **Akhil Kumar**

from Delhi Technological University has presented research paper on

Intrusion detection based on machine learning and statistical feature ranking techniques

during the 13th International Conference **Confluence 2023** on the theme **Cloud Computing, Data Science and Engineering** held on 19th - 20th January, 2023 at Amity University Uttar Pradesh, Noida.

Prof. (Dr.) Sanjeev Thakur
Conference Chair, Confluence-2023
HoD (CSE),
Amity School of Engineering & Technology
Amity University Uttar Pradesh, Noida, India

Prof. (Dr.) Abhay Bansal
General Chair, Confluence- 2023
Joint Head ASET, Director DICET,
Amity School of Engineering & Technology
Amity University Uttar Pradesh, Noida, India

Prof. (Dr.) Balvinder Shukla
Co-Patron, Confluence-2023
Vice Chancellor
Amity University Uttar Pradesh, Noida, India

PAPER NAME

final thesis akhil 1_removed.pdf

WORD COUNT

9812 Words

CHARACTER COUNT

54604 Characters

PAGE COUNT

50 Pages

FILE SIZE

900.5KB

SUBMISSION DATE

May 22, 2023 10:52 AM GMT+5:30

REPORT DATE

May 22, 2023 10:53 AM GMT+5:30

15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 10% Internet database
- 6% Publications database
- Crossref database
- Crossref Posted Content database
- 10% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 10 words)
- Manually excluded sources