HONEYPOT-BASED CYBER ATTACK DETECTION

A PROJECT WORK REPORT

Submitted by

ABISHEK PS (1901002)
HARI KISHORE V P (1901043)
KUGAANESEN (1901068)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SRI RAMAKRISHNA ENGINEERING COLLEGE

[Educational Service: SNR Sons Charitable Trust]
[Autonomous Institution, Re Accredited by NAAC with 'A+' Grade]
[Approved by AICTE and Permanently Affiliated to Anna University, Chennai]
[ISO 9001:2015 Certified and All Eligible Programmes Accredited by NBA]
Vattamalaipalayam, N.G.G.O. Colony Post,

COIMBATORE – 641 022

ANNA UNIVERSITY : CHENNAI 600 025

APRIL 2023

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

16CS270 –PROJECT WORK

Certified that this Project Work Report **"Honeypot-based cyber attack detection"** is the bonafide work of **"Abishek PS, Hari Kishore VP, Kugaanesen S"** who carried out the project under my supervision.

SIGNATURE

Mrs. M. Karthigha

SUPERVISOR

Assistant Professor (Sr. Grade),

Computer Science and Engineering,

Sri Ramakrishna Engineering College,

Coimbatore-641022.

SIGNATURE

Dr.Grace Selvarani

HEAD OF THE DEPARTMENT

Professor,

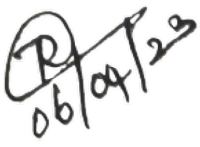
Computer Science and Engineering,

Sri Ramakrishna Engineering College,

Coimbatore-641022.

Submitted for the Project Work Viva-Voce Presentation held on 06-04-2023

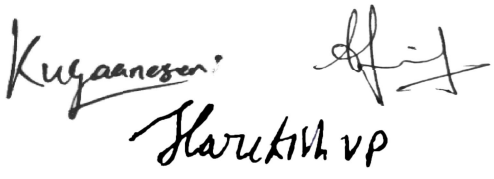
INTERNAL EXAMINER



EXTERNAL EXAMINER

DECLARATION

We affirm that the Project work titled **"HONEYPOT-BASED CYBER ATTACK DETECTION"** being submitted in partial fulfilment for the award of Bachelor of Engineering is the original work carried out by us. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.



(Signature of the Candidates)

ABISHEK PS (1901002)
HARI KISHORE VP (1901043)
KUGAANESEN S (1901068)

I certify that the declaration made above by the candidates is true.

(Signature of the guide)

Mrs. M. Karthigha,
Assistant Professor (Sr. Grade),
Department of CSE

ACKNOWLEDGEMENT

We express our gratitude to **Sri. D. LAKSHMINARAYANASWAMY,** Managing Trustee, **Sri. R. SUNDAR,** Joint managing Trustee, SNR Sons Charitable Trust, Coimbatore for providing excellent facilities to carry out our project.

We express our deepest gratitude to our Principal, **Dr. N. R. ALAMELU, Ph.D.,** for her valuable guidance and blessings.

We are indebted to our Head of the Department, **Dr. A. GRACE SELVARANI, Ph.D.,** Department of Computer Science and Engineering who modelled us both technically and morally for achieving great success in life.

We express our thanks to our Project Coordinator, **Mrs. S PRINCE SAHAYA BRIGHTY,** Assistant Professor (Sr. Grade) Department of Computer Science and Engineering for her great inspiration.

Words are inadequate to offer thanks to our respected guide. We wish to express our sincere thanks to **Mrs. M. KARTHIGHA,** Assistant Professor (Sr. Grade) Department of Computer Science and Engineering, who gives constant encouragement and support throughout this project work and who makes this project a successful one.

We also thank all the staff members and technicians of our Department for their help in making this project a successful one.

ABSTRACT

Cyber Intrusion is the most threatening expression in the cyber world, and it is a dangerous crime that many corporations and individuals who are a part of the Cyber World are horrified of. It results from not only a financial loss but also includes personal data which is impacted as a flooded river when the data is exposed in a data breach. Cyber Intrusion Detection System refers to a technology utilised for recognizing and notifying any illicit entry to a network system or network device. It analyses network traffic and logs files maintained by a device and reports or alerts when an outsider is trying to gain access to a network. Honeypot is a technology that acts as a catchy pot of honey for an attacker. When an attacker tries to catch the pot, the Honeypot system will alert the administrator and block it. Both technologies can be combined with Machine Learning to automate and improve the prediction rate so that attackers will be prevented. The use of Machine Learning algorithms detects the type of attacks. Further research should be conducted for the results to use the combination of honeypots, Cyber Intrusion, and Machine Learning to detect Cyber Attacks and the advancements, efficiency, and correctness of the prediction.

சுருக்கம்

சைபர் ஊடுருவல் என்பது சைபர் உலகில் மிகவும் அச்சுறுத்தும் வெளிப்பாடாகும், மேலும் இது சைபர் உலகின் ஒரு பகுதியாக இருக்கும் பல நிறுவனங்களும் தனிநபர்களும் திகிலடைவது ஆபத்தான குற்றமாகும். இது ஒரு நிதி இழப்பிலிருந்து விளைகிறது, ஆனால் தரவு மீறலில் தரவு வெளிப்படும் போது வெள்ளத்தில் மூழ்கிய நதியாக பாதிக்கப்படும் தனிப்பட்ட தரவையும் உள்ளடக்கியது. இணைய ஊடுருவல் கண்டறிதல் அமைப்பு என்பது பிணைய அமைப்பு அல்லது பிணைய சாதனத்தில் எந்தவொரு சட்டவிரோத நுழைவையும் அங்கீகரித்து அறிவிப்பதற்குப் பயன்படுத்தப்படும் தொழில்நுட்பத்தைக் குறிக்கிறது. இது நெட்வொர்க் ட்ராஃபிக்கை பகுப்பாய்வு செய்கிறது மற்றும் ஒரு சாதனத்தால் பராமரிக்கப்படும் கோப்புகளைப் பதிவு செய்கிறது மற்றும் வெளியாட்கள் நெட்வொர்க்கிற்கு அணுகலைப் பெற முயற்சிக்கும்போது அறிக்கைகள் அல்லது எச்சரிக்கைகள். ஹனிபாட் என்பது தாக்குதல் நடத்துபவருக்கு தேனின் கவர்ச்சியான பானையாக செயல்படும் தொழில்நுட்பமாகும். தாக்குபவர் பானையைப் பிடிக்க முயற்சிக்கும்போது, ஹனிபாட் அமைப்பு நிர்வாகியை எச்சரித்து அதைத் தடுக்கும். கணிப்பு விகிதத்தை தானியக்கமாக்குவதற்கும் மேம்படுத்துவதற்கும் இரண்டு தொழில்நுட்பங்களும் இயந்திர கற்றலுடன் இணைக்கப்படலாம், இதனால் தாக்குபவர்கள் தடுக்கப்படுவார்கள். இயந்திர கற்றல் அல்காரிதம்களின் பயன்பாடு தாக்குதல்களின் வகையைக் கண்டறியும். சைபர் தாக்குதல்களைக் கண்டறிய ஹனிபாட்கள், சைபர் ஊடுருவல் மற்றும் இயந்திர கற்றல் ஆகியவற்றின் கலவையைப் பயன்படுத்த முடிவுகளுக்கு மேலும் ஆராய்ச்சி நடத்தப்பட வேண்டும்.

TABLE OF CONTENTS

LIST OF FIGURES/TABLES

LIST OF ABBREVIATIONS

ML	**Machine Learning**
AI	**Artificial Intelligence**
GAN	**Generative adversarial networks**
IDS	**Intrusion Detection System**
DNS	**Domain Name System**
SVM	**Support Vector Machine**
KNN	**K - Nearest Neighbour**
DOS	**Denial of Service**
R2L	**Root to Local**
U2R	**User to Root**
IP	**Internet Protocol**
FTP	**File Transfer Protocol**
NIC	**Network Interface Card**
KDD	**Knowledge Discovery Database**
SMOTE	**Synthetic Minority Over-Sampling Technique**
NSL	**Network Security Laboratory**
DBIR	**Data Breach Investigation Report**

CHAPTER 1

INTRODUCTION

1.1 HONEYPOTS

The term "honeypot" refers to a trap designed to deceive cyber attackers by mimicking a real system or network. Honeypots are typically isolated from the production environment and are equipped with monitoring tools to record all the activities of the attackers. The captured data can then be used to analyse the attackers' methods and tactics and to develop effective countermeasures.

Honeypot-based cyber attack detection is a proactive approach to cybersecurity that allows organisations to gain valuable insight into the behaviour of attackers and their attack patterns. By analysing the data collected from the honeypot, cybersecurity experts can develop new security protocols and update existing ones to better defend against future attacks.

Moreover, honeypot-based detection can also be used as an early warning system, providing an indication that an attack is underway. This early warning can help organisations to take action quickly to mitigate the damage caused by the attack and reduce the time it takes to restore normal operations.

1.2 CYBER INTRUSION

The world is evolving with cutting-edge technology, and people around the world are interconnected with each other through the internet with the help of electronic devices and smart gadgets. There are about 5 billion active internet users worldwide[1]. A major threat to internet users is cyberattacks. Cyber-attacks may lead to any risk which includes financial losses, personal data leaks, business-related problems, corporate security, and mainly personal data security[8]. Data Breach sometimes lights up a few corporate secret crimes which are more

commonly found these days. Though it lights up crimes, Data Breach is a crime that involves one's or some personal details.

In the realm of cybersecurity, a honeypot serves as a mechanism to discover, divert, or counter any unauthorised exploitation of information systems. A honeypot is essentially a decoy system that is designed to attract and trap potential attackers by imitating a vulnerable system or application[7].

The idea behind a honeypot is to give attackers a fake system to attack instead of the real one, allowing security researchers to observe the attacker's techniques and tactics without risking damage to real systems. When an attacker interacts with a honeypot, the honeypot captures information about the attack, including the attacker's IP address, methods, and tools used in the attack. Among the several types of honeypots are high-interaction and low-interaction honeypots, with the former being designed to provide a realistic environment for attackers to operate in, whereas low-interaction honeypots simulate only certain aspects of a system. Honeypots can be used as a proactive security measure to both identify and thwart attacks, along with being a research tool to study and understand attackers' behaviour and motives.

1.3 CYBER THREATS

Cyber threats refer to any potential danger or risk posed by individuals, groups, or organisations that aim to compromise or exploit computer systems, networks, or devices for malicious purposes. Cyber threats can come in many forms, ranging from simple malware infections to complex and sophisticated cyber attacks. Some common types of cyber threats include:

- **Malware:** Malware is a type of software that is designed to damage, disrupt, or gain unauthorised access to a computer system or network. Examples of malware include viruses, worms, and Trojan horses.

- **Phishing:** Phishing is a social engineering technique that uses fraudulent emails, text messages, or websites to trick individuals into disclosing sensitive information such as login credentials or credit card numbers.
- **Denial of Service (DoS) attacks:** DoS attacks are designed to overwhelm a website or network with traffic, making it unavailable to legitimate users.
- **Advanced Persistent Threats (APTs):** APTs are sophisticated and targeted attacks that are designed to gain long-term access to a network or system.
- **Ransomware:** Ransomware is a type of malware that encrypts files on a computer system and demands payment in exchange for the decryption key.
- **Insider threats:** Insider threats refer to individuals within an organisation who use their access to systems and information for malicious purposes, such as stealing sensitive data or sabotaging operations.

These are just a few examples of the many types of cyber threats that exist. As technology continues to evolve and become more interconnected, the threat landscape is also constantly evolving, making it increasingly important for individuals and organisations to be aware of the latest threats and take steps to protect themselves against them.

The global average cost of a Data breach is about 4.35 USD[14]. Verizon's 2022 Data Breach Investigation Report (DBIR) states that about 62% of incidents are of System Intrusion patterns involving threat actors compromising partners. 13% increase in Ransomware which is more than the combined past 5 years[13]. Avoiding such attacks for an individual is impossible since targeted attacks are more common these days where the data of high-powered people are not safe. To avoid such attacks in a network, the use of new technologies is a must to get more secure. The attackers are smarter as there are a few ways to bypass the honeypots.

Data can be secured in many ways, but the most efficient in terms of power consumption, data consumption, and quick response is more important. The combination of Honeypot and Cyber Intrusion Detection ways using Machine Learning Algorithms can predict almost every input data frame in less time with a high prediction rate which can stop the attacker from accessing the data.

In the realm of cybersecurity, there is a constant need for new and effective methods to detect and prevent cyber attacks. One such method is honeypot-based cyber attack detection, which involves setting up a decoy system or network to lure cyber attackers and then monitoring their activities to gain insights into their tactics, techniques, and procedures.

1.4 INTRUSION DETECTION

Intrusion detection is the process of monitoring a computer network or system to detect unauthorised access or activity. The goal of intrusion detection is to identify and respond to security incidents in a timely and effective manner, in order to prevent or minimise damage to the network or system.

There are two main types of intrusion detection: host-based and network-based. Host-based intrusion detection involves monitoring activity on individual systems or devices, while network-based intrusion detection involves analysing network traffic to identify suspicious activity.

Intrusion detection systems (IDS) can be either rule-based or anomaly-based. Rule-based IDS use a set of predefined rules or signatures to identify known attacks or patterns of suspicious activity. Anomaly-based IDS, on the other hand, use machine learning or statistical techniques to identify deviations from normal patterns of activity, which may indicate an attack.

Intrusion detection can be further categorised as either passive or active. Passive intrusion detection involves monitoring activity and generating alerts when

suspicious activity is detected, but does not take any action to prevent or stop the activity. Active intrusion detection, on the other hand, may take action to block or prevent suspicious activity, such as by closing network ports or blocking IP addresses.

In summary, intrusion detection is a crucial aspect of network security that involves monitoring network activity to detect unauthorised access or suspicious activity. Intrusion detection systems can be rule-based or anomaly-based, and can be either passive or active. While intrusion detection is not perfect, it is an important tool for preventing and responding to security incidents, and should be an essential component of any comprehensive network security strategy.

CHAPTER 2

LITERATURE REVIEW

2.1 STUDY OF THE ATTRIBUTES USING FOUR CLASS LABELS ON KDD99 AND NSL-KDD DATASETS WITH MACHINE LEARNING TECHNIQUES *(Nilesh Kunhare and Ritu Tiwari)*

The paper begins by providing a brief overview of the KDD99 and NSL-KDD datasets, which are widely used in the field of network intrusion detection. The authors then describe their methodology for preprocessing the datasets and extracting relevant features, such as the number of packets sent and received, the duration of the connection, and the protocol used.

Next, the authors compare the performance of different machine learning algorithms on the datasets, including decision trees, k-nearest neighbors (KNN), support vector machines (SVM), and neural networks. The performance is evaluated based on several metrics, such as accuracy, precision, recall, and F1-score.

The authors find that different machine learning techniques perform differently on the datasets depending on the class label being predicted. For example, SVM and neural networks perform well for detecting DOS attacks, while decision trees and KNN are more effective for detecting R2L attacks. Additionally, the authors note that the NSL-KDD dataset generally performs better than the KDD99 dataset, likely due to the former's inclusion of more recent attack types.

The paper also includes a discussion of the most important features for each class label, as identified by the machine learning algorithms. For example, the authors find that the number of failed login attempts is an important feature for

detecting R2L attacks, while the number of outgoing packets is a significant feature for detecting probe attacks.

The authors conclude by summarising their findings and highlighting the importance of selecting appropriate machine learning techniques for the specific task of intrusion detection. They also emphasise the need for further research to improve the accuracy and effectiveness of intrusion detection systems.

Overall, the paper provides a comprehensive analysis of the effectiveness of different machine learning techniques for detecting attacks in the KDD99 and NSL-KDD datasets. The findings are valuable for researchers and practitioners in the field of network security, and can inform the development of more accurate and effective intrusion detection systems.

2.2 AI POWERED NETWORK THREAT DETECTION SYSTEM *(Bo-Xiang Wang and Jiann-Liang Chen)*

The system consists of several modules, including data preprocessing, feature extraction, and classification. In the data preprocessing module, raw network traffic data is transformed into a format suitable for analysis. In the feature extraction module, relevant features are extracted from the data using statistical and time series analysis techniques. The authors also propose a novel feature selection method based on the correlation between features and the target variable.

The classification module uses a deep neural network (DNN) to classify network traffic as normal or anomalous. The authors propose a new type of DNN architecture called Deep Sparse Autoencoder (DSA), which incorporates both supervised and unsupervised learning. The supervised learning component uses labelled data to train the DNN to classify network traffic, while the unsupervised learning component helps to identify unknown threats and anomalies.

The authors evaluate their system on several benchmark datasets, including the UNSW-NB15 dataset and the CIC-IDS2017 dataset. They compare the performance of their system to several other state-of-the-art approaches and demonstrate that their system achieves high accuracy and detection rates. They also perform a sensitivity analysis to evaluate the impact of different parameters on the performance of the system.

Overall, the paper presents a promising approach to network threat detection using machine learning and deep learning techniques. The proposed system combines supervised and unsupervised learning to improve the accuracy and efficiency of threat detection, and the results demonstrate that it outperforms other state-of-the-art approaches on benchmark datasets. The authors suggest that their system could be used to improve the security of enterprise networks and other critical infrastructure.

2.3 LIMITATIONS

While cyber intrusion detection systems (IDS) can be effective in detecting and preventing cyber attacks, they are not without limitations. Some of the key limitations include:

- **False positives:** IDS can sometimes generate false positives, which occur when an alert is triggered for a non-malicious activity. This can lead to unnecessary alerts and a waste of resources.
- **Zero-day attacks:** IDS may not be able to detect zero-day attacks, which are attacks that exploit vulnerabilities that are unknown to the security community. These attacks can be particularly challenging to detect and prevent.

- **Advanced persistent threats (APTs):** APTs are a type of cyber attack that are specifically designed to evade detection by security systems. They can be difficult to detect and may require specialised tools and techniques.
- **Encryption:** Encrypted traffic can be difficult to inspect for signs of malicious activity. Attackers may use encryption to hide their activities from IDS and other security systems.
- **Cost:** Implementing and maintaining an IDS can be expensive, requiring specialised hardware and software, as well as trained personnel to monitor and respond to alerts.

Overall, while IDS can be an important component of a comprehensive cyber security strategy, they are not a silver bullet solution and must be supplemented by other security measures to effectively detect and prevent cyber attacks.

CHAPTER 3

SYSTEM DESCRIPTION

3.1 HARDWARE REQUIREMENTS

1. Computer:

   system that is capable of handling a wide range of tasks. The 2.5GHz processor can execute 2.5 billion cycles per second, allowing for fast and efficient processing of data. The 16GB of RAM provides ample memory for multitasking and running memory-intensive applications. The 4GB GPU is capable of handling complex graphics processing tasks, such as video editing or gaming. Overall, this computer configuration is well-suited for a range of applications, mainly for Machine Learning.

2. Network Interface Card:

   A Network Interface Card (NIC) is a hardware component that provides connectivity between a computer and a network. It is typically installed in a computer's expansion slot or built into the motherboard, and allows the computer to communicate with other devices on the network using a variety of communication protocols, such as Ethernet or Wi-Fi. The NIC is responsible for transmitting and receiving data packets over the network, and can also handle tasks such as error checking and packet filtering. NICs can vary in speed, from 10 Mbps to 100 Gbps or higher, and can also include additional features such as Wake-on-LAN and VLAN tagging.

3.2 SOFTWARE REQUIREMENTS

1. Virtual Machine:
    - **VMWare Workstation Player 17**

A desktop virtualization software that allows users to run multiple operating systems simultaneously on a single computer. It provides a sandbox environment for testing software and running legacy applications without interfering with the host operating system. The software supports a wide range of operating systems, including Windows, Linux, and macOS, and provides advanced features such as virtual networking, snapshotting, and remote connections. VMware Workstation Player 17 is designed for personal use and is free for non-commercial use, making it a popular choice among students, developers, and IT professionals.

2. Operating System:
    - **Windows (Host)**
    - **Kali Linux (Victim Machine)**
    - **Metasploit Linux (Attack Machine)**

This configuration describes a typical penetration testing setup, where a Windows host is used to launch attacks on a Kali Linux victim machine using the Metasploit framework running on a separate Linux machine. Kali Linux is a popular distribution for penetration testing and includes a wide range of tools for testing the security of networks and systems. Metasploit is a powerful tool that allows users to create and launch exploits against vulnerabilities in target systems. By using a separate attack machine, the user can isolate the risks associated with launching attacks and better control the testing environment. This setup is commonly used in both ethical hacking and security testing to assess the security of networks and systems.

3. Programming Language:
    - **Python3 (ML Training)**
    - **C++ (Feature Extraction from Network Traffic)**

Python3 is a popular programming language used for machine learning training and development. It has a variety of libraries and frameworks that make it easy to create and train machine learning models for a range of applications.

In the context of network traffic, C++ can be used to extract features such as packet size, time between packets, and other metadata that can be used to identify patterns and anomalies in the traffic.

4. Network Monitor:
    - WireShark 4

Wireshark is a popular network protocol analyzer that allows users to capture and analyse network traffic in real-time. It supports a wide range of protocols and can be used to troubleshoot network issues, identify security threats, and analyse network performance. Wireshark provides a user-friendly interface and allows users to filter and search through captured packets to quickly identify specific traffic patterns or events. It is open source software and is widely used by network administrators, security analysts, and developers for network analysis and debugging.

5. Code Editor
    - **Visual Studio Code**

Visual Studio Code, often abbreviated as VS Code, is a free, open-source code editor developed by Microsoft. It supports a variety of programming languages and provides a range of features such as syntax highlighting, debugging, and Git integration.

CHAPTER 4

MODELLING ATTRIBUTES

4.1 NETWORK DESIGN

The network design is designed in such a way that all the data is the hidden backside of a honeypot server divided by firewalls. Multiple layered designs are required to attract more attackers and it helps in identifying attackers which use honeypot bypass techniques. It is explained well in the following figure with network design designed using Cisco Packet Tracer.

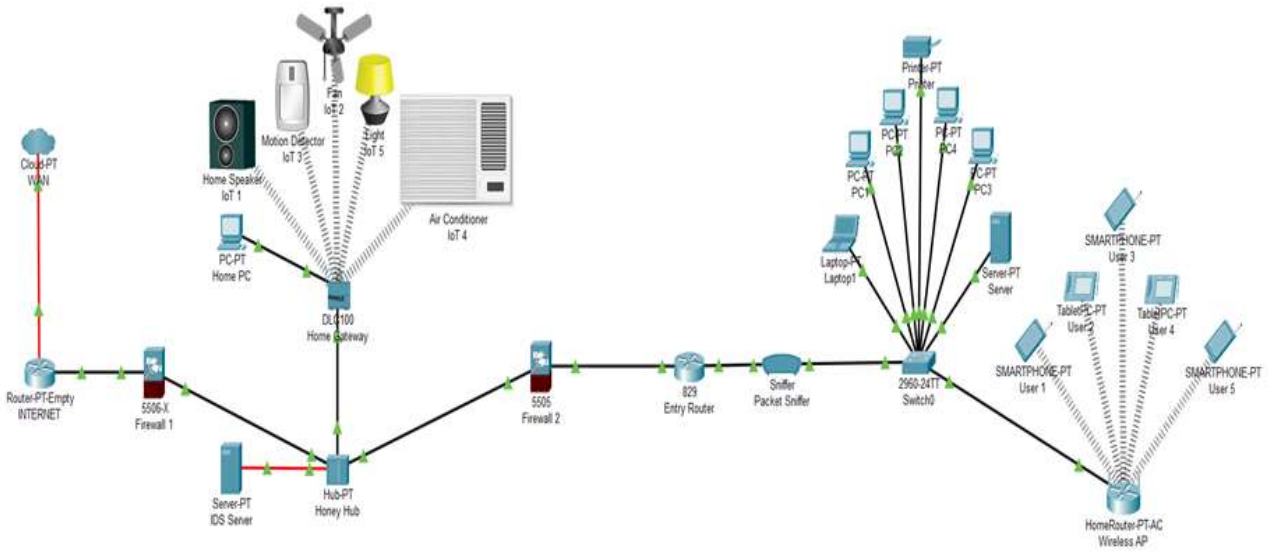


Fig. 4.1 Network Design

I. The whole network is entered directly through the Internet router from the Internet Service Provider (ISP) or the outer network.

II. All the traffic from the Internet Service Provider enters the Internet Router and the data packets are sent over Firewall 1 to filter the usual unwanted packets from the internet or the outer network.

III. Filtered packets then enter the Honey Hub. The honey Hub is connected to a home gateway that is fully connected with more devices using the internet (IoT Devices), this may be physical devices or may be virtual devices running old or vulnerable operating systems. Old and vulnerable software versions for IoT devices are the 1st layer of trap for attackers. Since the software is vulnerable by nature, they are more likely to act as a natural honeypot.

IV. The role of the Honey HUB is to transmit all packets to the IDS server since HUB broadcasts every packet detail to every device connected to it.

V. The honeypot server which is connected via optical fibre receives data more quickly since they are much faster than the normal twisted pair of Cat 5, Cat 6, Cat 7, and Cat7E cables.

VI. The IDS server analyses the network and uses Machine Learning Algorithm that predicts the packet's nature and collects all required details for the prediction and sends the report to Firewall 2 via Honey Hub with an encrypted secured channel.

VII. Entry Router receives packets that are filtered by Firewall 2 as directed by the honeypot server. The packets move via a packet sniffer which collects all data for future study purposes since “Nothing on the internet is 100% safe”.

Finally, the Switch0 and the Home Router or the regular network are connected and can be used normally.

4.2 HONEYPOT CREATION

A honeypot is a security mechanism that can be used to detect and analyse attempted unauthorised access to a computer system or network. The process of creating a honeypot involves setting up a system or network that appears to be vulnerable to attacks, with the purpose of luring attackers and gathering information about their methods and techniques.

To create a honeypot, the type of honeypot that is to be deployed needs to be selected. Several types of honeypots are available, such as high-interaction, low-interaction, and medium-interaction honeypots, and each type has its advantages and disadvantages, depending on the level of risk that is acceptable.

Next, the architecture of the honeypot system needs to be designed. Factors to consider include the type of operating system, the network topology, and the services that are to be run on the honeypot system. It is also essential to consider the security measures that need to be implemented to protect the honeypot system from attackers.

Once the architecture has been designed, the honeypot system can be implemented. This involves installing the operating system, configuring the network, and installing the necessary services. Monitoring tools should also be set up to collect data on the attackers' activities.

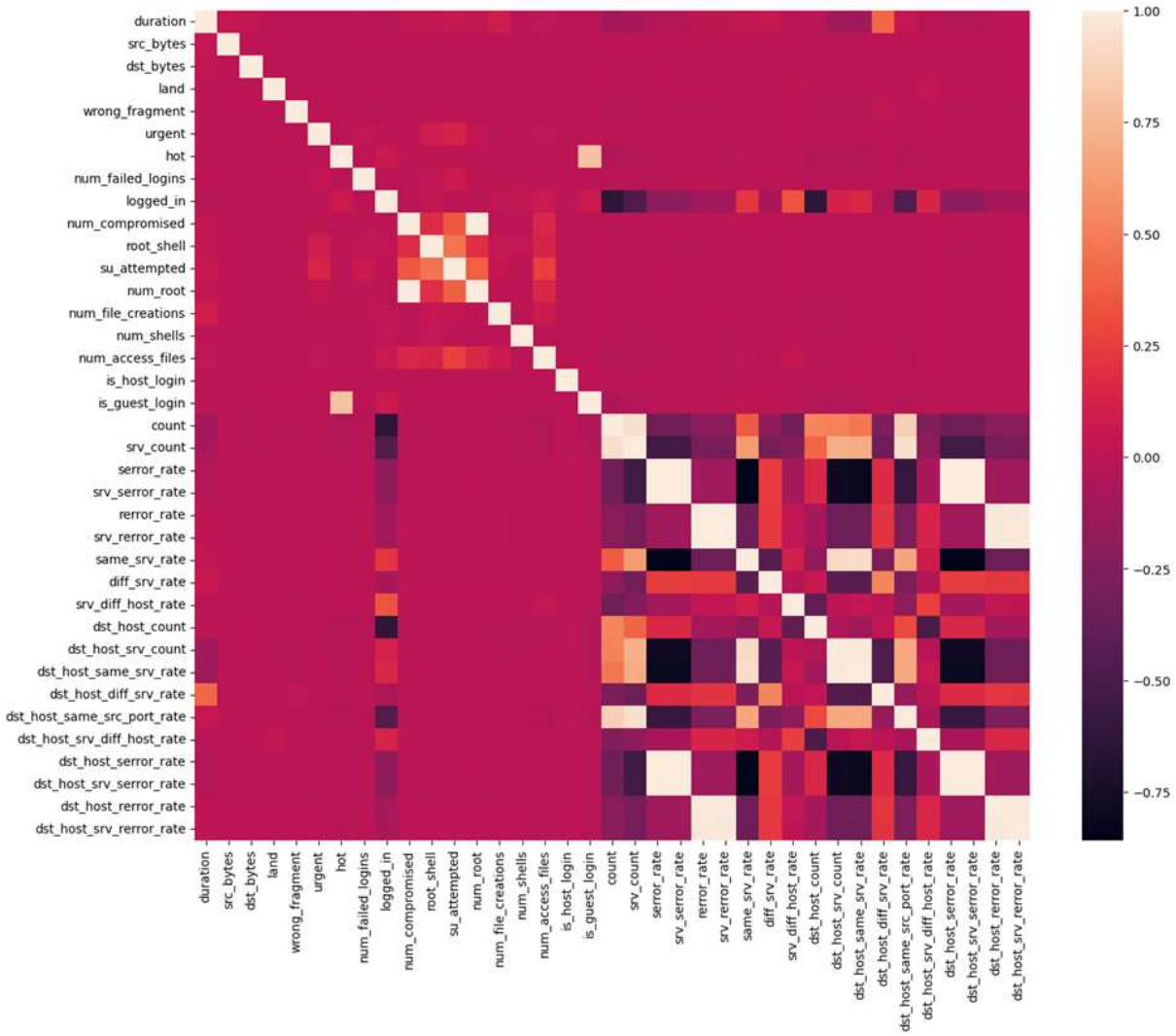
Finally, the data collected by the honeypot system needs to be analysed to gain insights into the attackers' tactics and motives. This can help improve the organisation's security posture and protect against future attacks.

4.3 DATASET

The dataset used is the one released by The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining. The task was to build a network intrusion detector, a predictive model capable of distinguishing between “bad'” connections, called intrusions or attacks, and “good” normal connections. The dataset has 42 columns which are the main features that determine the prediction. The most important field mainly consists of attack type and type of connection are as follows:

- ‘Normal’ for regular data.
- ‘dos’ type of attack for ‘Neptune, smurf, land, back, pod, teardrop’.
- ‘probe’ type of attack for the type ‘ipsweep, satan, port sweep or nmap’.
- ‘r2l’ type of attack for type ‘ftp_write, guess_passwd, imap, phf, spy, warezclient, warezmaster or multihop’
- ‘u2r’ for ‘Perl, loadmodule, buffer_overflow or rootkit’

The database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, but the combination of honeypots with a 2 layered intrusion model using different ML Algorithms helps in identifying more efficiently and with low processing power. The following figure 4.2 shows the heatmap of all the features of the KDD99 dataset for all 42 features.



Fig, 4.2 KDD99 Heatmap

Data preprocessing is an essential step in using the KDD99 dataset for intrusion detection research. The dataset contains a large amount of raw network traffic data, which needs to be processed and transformed into a format that can be used for machine learning algorithms. The first step in preprocessing the data is to remove any duplicates or irrelevant data. The KDD99 dataset contains duplicate records and some records that do not provide useful information for intrusion detection, such as connection records with zero duration or invalid values.

Next, the data needs to be transformed into a suitable format for machine learning algorithms. This includes converting categorical data, such as protocol types and service names, into numerical values using one-hot encoding. Features that are highly correlated or do not contribute much to the classification task can be removed to reduce dimensionality. To ensure fairness in the evaluation of IDS algorithms, the KDD99 dataset is typically split into training, validation, and testing sets. The training set is used to train the model, while the validation set is used to tune hyperparameters and prevent overfitting. The testing set is used to evaluate the performance of the model on unseen data.

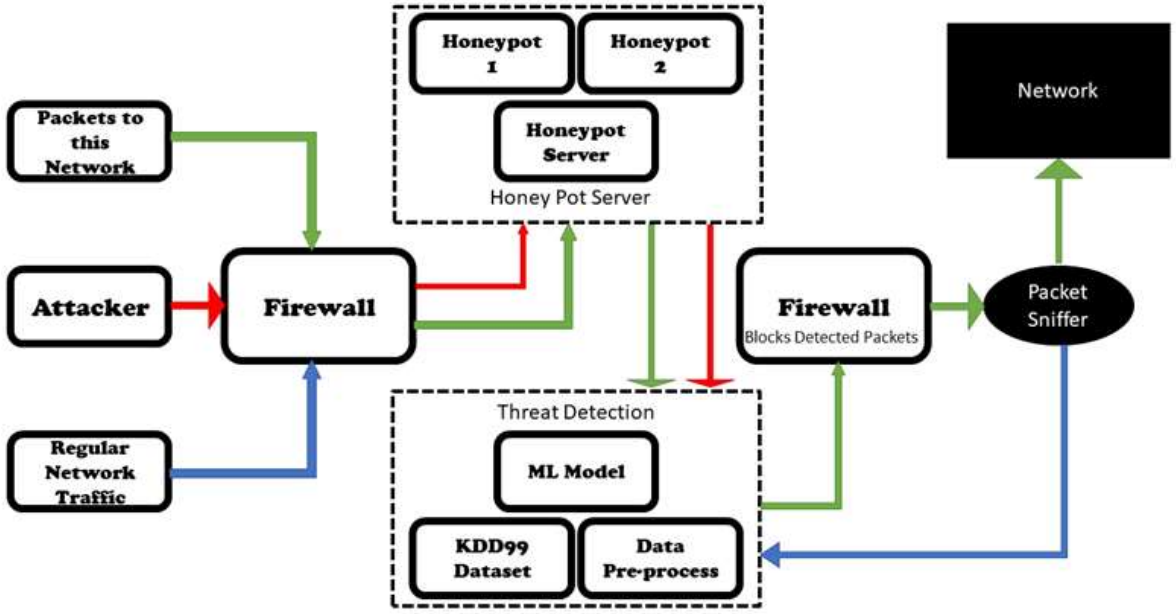
Finally, it is important to balance the dataset to ensure that there is an equal representation of normal and attack traffic. This can be achieved by either undersampling the majority class or oversampling the minority class using techniques such as random oversampling or SMOTE (Synthetic Minority Over-sampling Technique).

Overall, data preprocessing is a critical step in using the KDD99 dataset for intrusion detection research and can have a significant impact on the performance of the IDS algorithms.

4.4 INTRUSION DETECTION SYSTEM BY MACHINE LEARNING

The common packets which are filtered by Firewall are almost safe from regular intrusions and regular web traffic. The firewall may be a hardware firewall or else a software-based firewall to filter the packets. All the data together come to the HUB and are broadcasted to every device. A smart attacker can sense the use or any other mode of packet sniffing, and even the intruder can modify or destroy log files and reports generated by the Honeypot Server, but the use of Hub makes it look like a natural device used for connecting devices. The possibility of identifying IoT devices is very high because of the vulnerable software running in

them. All the data is monitored by software or hardware in the Honeypot server. The figure 4.3 explains well as a block diagram



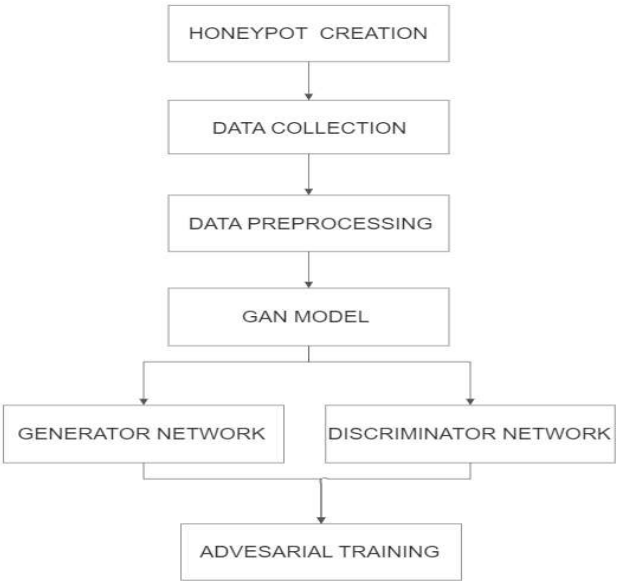
Fig, 4.3 Block diagram

Honeypot Server is the main part that determines whether the packets are safe or not. It analyses every single packet in detail and gives predictions using Machine Learning Algorithm. Every machine learning model has its specific type of processing method. Every classifier has its working algorithms. As the desired output is a prediction of type “intruder” or “Normal” type, a proper machine learning model is to be selected for the required output. A set of ML classifiers are taken to test the dataset. The dataset has a label column which is suitable for supervised machine-learning algorithms. The machine learning algorithms used are Gaussian Naive Bayes, Decision Tree, Random Forest, Gradient Boosting, Extreme Gradient Boosting, Generative adversarial network, Linear Regression, Logistic Regression, Long Short-Term Memory, and Deep Belief Network.

Different Machine Learning algorithms are used to determine whether the connection is good or it is an intruder. The data from the honeypot is analysed well by the Machine Learning model which has about 42 features used for prediction. There are tested using various Machine Learning models such as Gaussian Naive Bayes, Decision Tree, Random Forest, Gradient Boosting, Extreme Gradient Boosting, Generative adversarial network, Linear Regression, Logistic Regression, Long Short-Term Memory, and Deep Belief Network as mentioned above. The one which is suitable for intrusion detection gives a high accuracy value and the high accuracy model is selected for the detection.

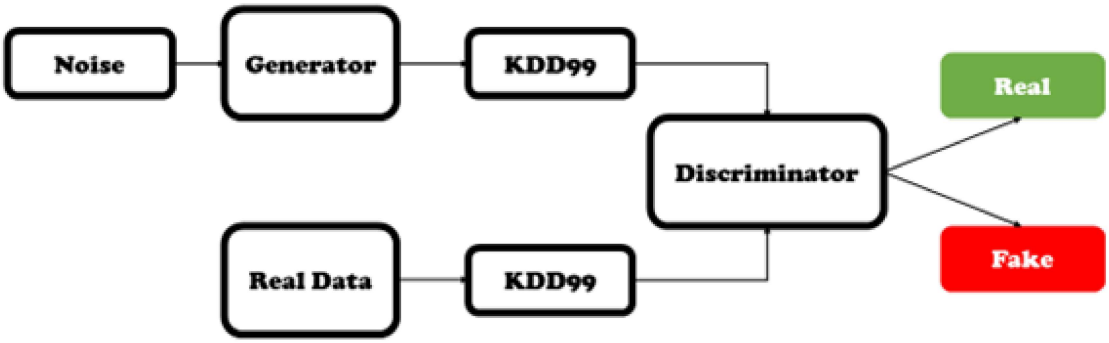
Generative Adversarial Networks (GANs) are a class of deep learning algorithms used to generate new and synthetic data by pitting two neural networks against each other in a zero-sum game framework.

There are two main components: a generator and a discriminator. The generator takes a random noise vector as input and maps it to an output that is meant to resemble the target data distribution. The discriminator takes in both real samples from the target data distribution and fake samples generated by the



Fig, 4.4 Flow diagram

generator and tries to distinguish between the two, they are trained in an adversarial manner, generator creating samples that are realistic to the discriminator, the discriminator tries to classify the samples as either real or fake. Over time, the generator improves its ability to create realistic samples, and the discriminator improves its ability to identify fake samples.



Fig, 4.5 GAN Block Diagram

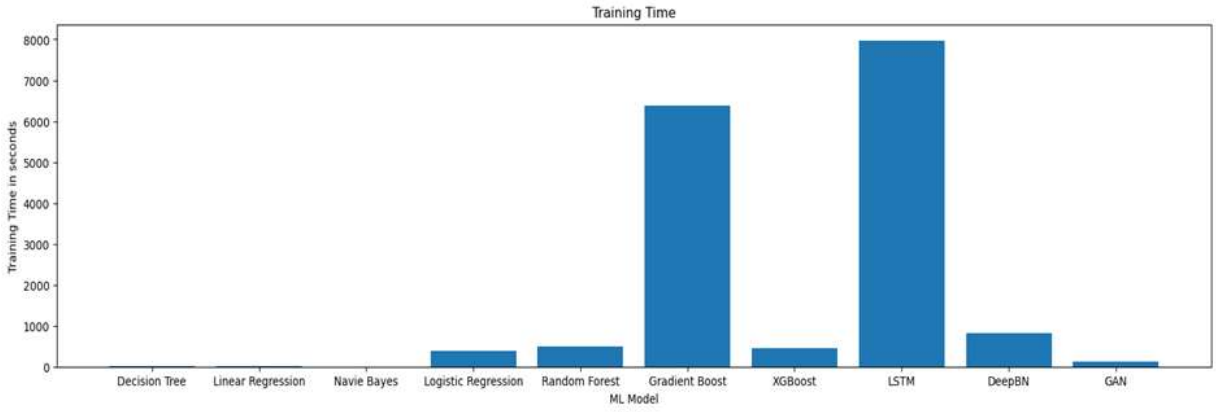
Since all the classifiers perform well for the used dataset, we need to consider some other factors as the prediction is used for security purposes and threat detection. The dataset has 48,98,430 columns and 42 Rows with different features. As the predictions are used in security and threat detection applications, we need to take care more of the input dataset. Generative Adversarial Networks can be used if their accuracy is good enough compared to other classifiers.

4.5 GAN VS OTHER MODELS

Generative adversarial networks (GANs) are a type of machine learning algorithm that have shown great success in generating synthetic data that closely resembles real-world data. They consist of two neural networks, a generator and a

discriminator, that are trained in a competitive process to produce increasingly realistic synthetic data.

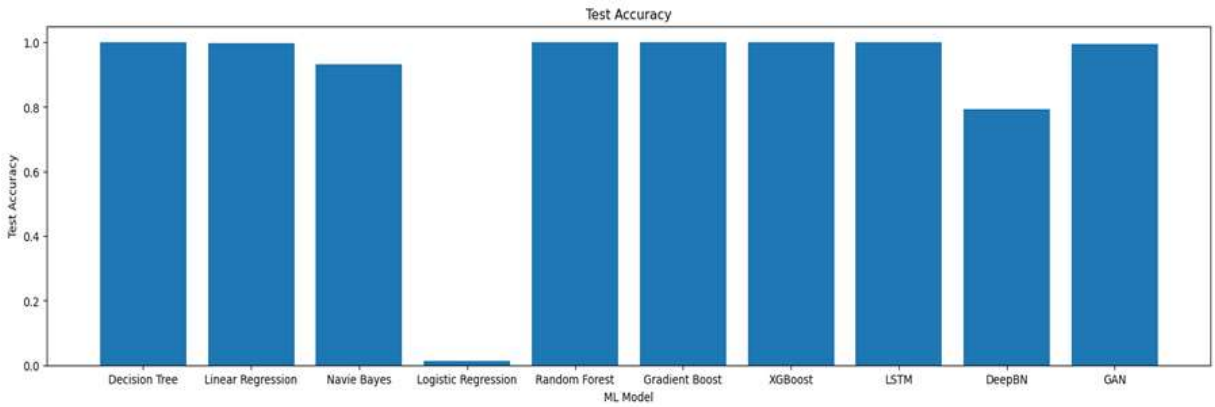
While the traditional machine learning algorithms such as Gaussian Naive Bayes, Decision Tree, Random Forest, Gradient Boosting, Extreme Gradient Boosting, Linear Regression, Logistic Regression, Long Short-Term Memory, and Deep Belief Network have been extensively used for cyber threat detection, they are typically limited in their ability to accurately identify complex and evolving threats.



Fig, 4.6 Training Time (in seconds)

On the other hand, GANs have shown great potential in generating synthetic data that closely resembles real-world data, which can be used to improve the accuracy of cyber threat detection models. For example, GANs can be used to generate synthetic network traffic that includes a wide range of known and unknown attack scenarios, which can then be used to train cyber threat detection models.

Moreover, GANs are capable of learning from unlabeled data, making them suitable for detecting novel and emerging threats that may not have been previously identified. This ability to adapt and learn from new data is critical in the rapidly evolving field of cyber threat detection.



Fig, 4.7 Testing Accuracy

Therefore, while traditional machine learning algorithms may have their applications in cyber threat detection, GANs offer significant advantages in terms of accuracy, adaptability, and the ability to generate synthetic data for training and testing purposes.

CHAPTER 5

RESULTS

Realtime Honeypots capture intruders by various methods and the data required for the prediction is generated by it. As a whole, it is passed to the various Machine Learning models to get its performance and accuracy. Test and train time is also considerable for power-efficient working.

The Decision Tree algorithm shows a test accuracy of 99.62% and a training time of 10.85s. The Linear Regression algorithm shows a test accuracy of 97.75% and a training time of 5.99s. The Gaussian Naive Bayes algorithm shows a test accuracy of 93.21% and a training time of 3.74s. The Logistic Regression algorithm shows a test accuracy of 1.41% and a training time of 376.92s. The Random Forest algorithm shows a test accuracy of 99.92% and a training time of 495.71s. The Gradient Boosting algorithm shows a test accuracy of 95.94% and a training time of 6372.18s. The Extreme Gradient Boosting algorithm shows a test accuracy of 98.93% and a training time of 443.02s. The Long Short-Term Memory algorithm shows a test accuracy of 97.87% and a training time of 7966.02s. The Deep Belief Networks algorithm shows a test accuracy of 79.84% and a training time of 819.10s. The Generative Adversarial Networks algorithm shows a test accuracy of 99.99% and a training time of 130.02s. The table 5.1 shows the results of all the classifiers and its training time with testing accuracy of each.

Classifier	Test Accuracy	Training Time
Decision Tree	0.9962	10.846
Linear Regression	0.9775	5.988
Navie Bayes	0.9321	3.739
Logistic Regression	0.0141	376.921
Random Forest	0.9992	495.712
Gradient Boost	0.9594	6372.185
XG Boost	0.9893	443.016
LSTM	0.9787	7966.016
Deep BN	0.7984	819.102
GAN	0.9999	130.016

Table, 5.1 Classifiers Performance

Epoch	Discriminator Loss	Generator Loss	Accuracy
1	38.05141	0.689847	42.19
1000	0.131214	1.496475	100
2000	0.096201	1.761345	100
3000	0.025205	3.041839	100
4000	0.011418	3.800014	100
5000	0.005777	4.490452	100
6000	0.00349	4.992599	100
7000	0.002289	5.401649	100
8000	0.001711	5.725489	100
9000	0.001114	6.107335	100
10000	0.000896	6.343067	100

Table, 5.2 GAN Results

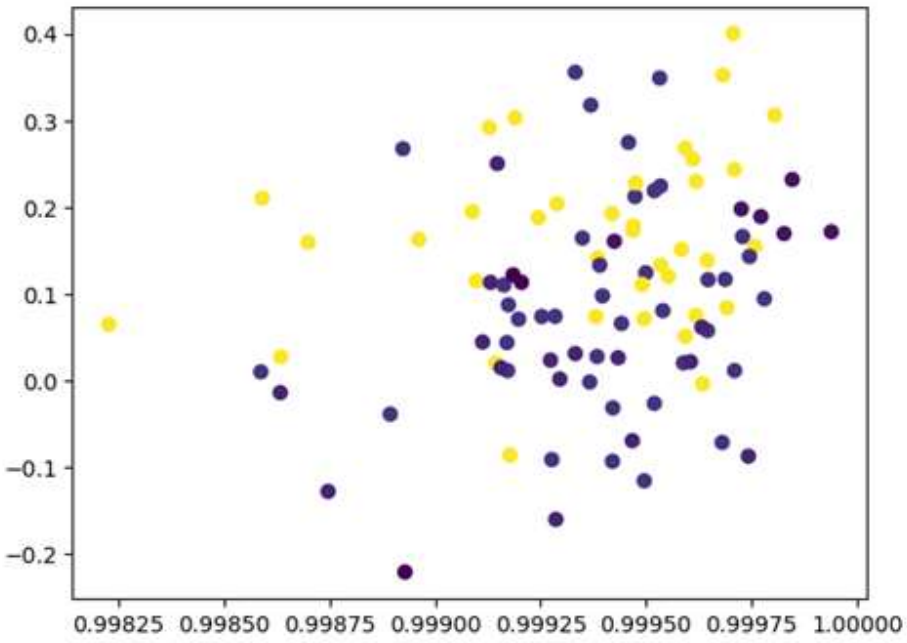
The graph 3.1 and 3.2 specifies the testing accuracy and the training time of the different classifiers as bar graphs. In testing accuracy, Logistic Regression has performed the least and almost all other classifiers score more than 80. The bar graph says more about the training time required by all the classifiers.

CHAPTER 6

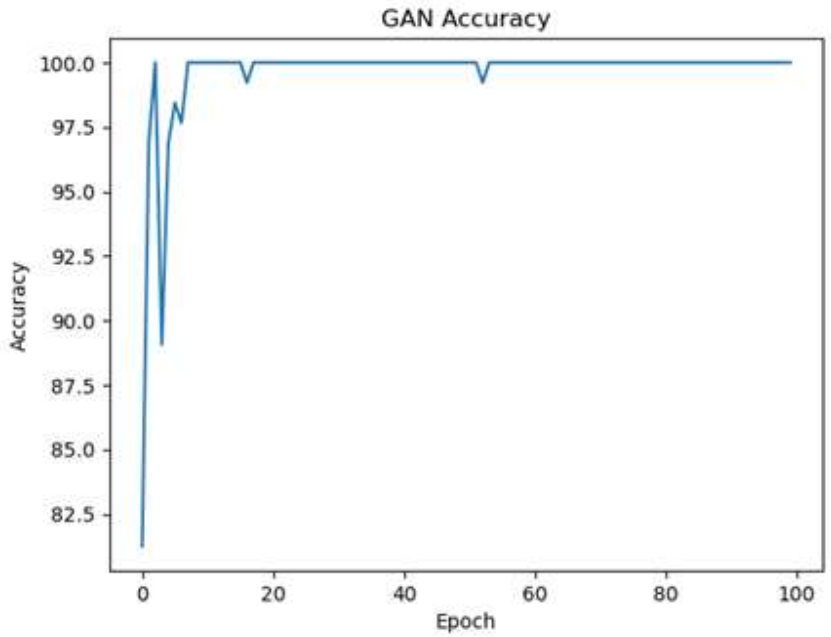
CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

The use of Generative Adversarial Networks is used for intrusion detection. The maximum-performing algorithm exists but it is limited by the size of the dataset. Since Generative Adversarial Networks are trained using synthetically generated values, their prediction and accuracy are still high. Other models have been trained with just the input dataset of size 48,98,430 columns. The GAN gives accurate results, compared to all other algorithms GAN is trained with a high number of inputs with about 1000 epochs. The use of network design is to reduce the regular traffic and to catch the attackers. Future research should verify the new methods of attackers and include more features for advanced attacks.



Fig, 6.1 GAN Generated Data



Fig, 6.2 GAN Accuracy

The Generative Adversarial Network gives the most accurate result for the KDD99 dataset with almost more accuracy that most of the accuracy plot in the graph falls within 1 and 0.99825 as shown in the figure 4.1 and figure 4.2. Since synthetic data is used to train it, it is more accurate than any classifiers used for the same dataset. Thus, Generative adversarial network has been selected for the Intrusion Detection System.

6.2 FUTURE SCOPE

The use of honeypots and generative adversarial networks (GANs) for cyber attack detection is an emerging field that holds great promise for improving the security of networks and systems. As this approach is relatively new, there is still significant potential for future research and development in this area.

One potential future direction is the development of more sophisticated GAN models that can generate synthetic data with even greater accuracy and fidelity, thereby improving the effectiveness of intrusion detection systems.

Another area for future research is the application of GAN-based cyber attack detection to other types of networks and systems, such as industrial control systems and critical infrastructure.

In addition, further investigation is needed to better understand the limitations and potential biases of GAN-based intrusion detection systems, particularly with regard to the potential for false positives and false negatives. Improved methods for evaluating the performance of GAN-based systems, such as the use of more realistic and representative datasets, can also help advance the field.

Finally, there is potential for the integration of GAN-based intrusion detection systems with other cybersecurity technologies, such as anomaly detection and machine learning-based threat intelligence, to create more comprehensive and effective defence systems. Overall, the use of GANs in combination with honeypots offers significant potential for improving cyber attack detection and defence, and continued research and development in this area is likely to yield significant benefits in the future.

6.2.1 MERITS AND DEMERITS

Merits:

GAN can generate synthetic attack data, which can be used to train intrusion detection systems (IDS) in a more realistic and diverse manner, improving their ability to detect new and sophisticated attacks. Honeypots can be used to gather information about attackers' behaviour and tactics, enabling organisations to improve their security posture and develop new countermeasures. Honeypots can be deployed in a targeted manner to lure attackers to specific areas of the network or systems, allowing organisations to monitor and track their activities more closely. The use of GAN and honeypots together can help address the problem of

imbalanced datasets in IDS, which is a common issue in traditional ML-based approaches.

Demerits:

Honeypots can be difficult to deploy and maintain, requiring significant resources and expertise to implement effectively. The use of honeypots can also pose a potential security risk if not properly secured, as attackers may be able to use them as a foothold to launch attacks on the organisation's actual systems and network. GAN-generated data may not always accurately reflect real-world attack scenarios, as the synthetic data may be biassed or incomplete in certain areas.GAN-based approaches can be computationally intensive, requiring significant resources and time to generate and analyse synthetic data, which may limit their scalability and practicality in certain contexts. Overall, while honeypot-based cyber attack detection using GAN has several benefits, it also requires careful consideration of its potential limitations and risks.

CHAPTER 7

APPENDICES

7.1 SAMPLE CODE

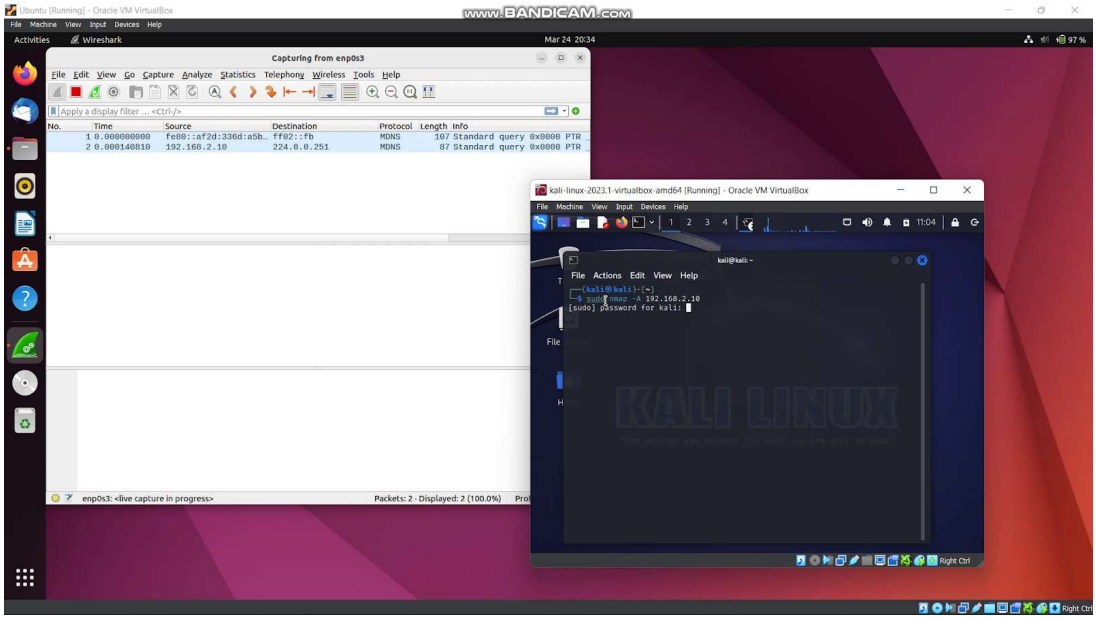
```
import numpy as np
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
# Load the KDD99 dataset
dataset = pd.read_csv("../dataset/kddcup.data/data.csv")
# Preprocess the data
def preprocess(df):
    categorical_columns = [col for col in df.columns if df[col].dtype ==
'object']
    for col in categorical_columns:
        df[col] = df[col].astype('category').cat.codes
    # df.to_csv("../Preprocess/kddcup.data.csv")
    return df.values

df = preprocess(dataset)
# Split the data into training and test sets
train_data = df[:int(len(df) * 0.9)]
test_data = df[int(len(df) * 0.9):]
# Define the generator model
def make_generator_model():
    model = tf.keras.Sequential()
    model.add(tf.keras.layers.Dense(256, input_dim=100))
    model.add(tf.keras.layers.LeakyReLU())
    model.add(tf.keras.layers.BatchNormalization(momentum=0.8))
    model.add(tf.keras.layers.Dense(512))
    model.add(tf.keras.layers.LeakyReLU())
    model.add(tf.keras.layers.BatchNormalization(momentum=0.8))
    model.add(tf.keras.layers.Dense(1024))
    model.add(tf.keras.layers.LeakyReLU())
    model.add(tf.keras.layers.BatchNormalization(momentum=0.8))
    model.add(tf.keras.layers.Dense(np.prod(train_data.shape[1:]),
activation='tanh'))
    model.add(tf.keras.layers.Reshape(target_shape=train_data.shape[1:]))
    return model
```

```
generator = make_generator_model()
# Define the discriminator model
def make_discriminator_model():
    model = tf.keras.Sequential()
    model.add(tf.keras.layers.Flatten(input_shape=train_data.shape[1:]))
    model.add(tf.keras.layers.Dense(1024))
    model.add(tf.keras.layers.LeakyReLU())
    model.add(tf.keras.layers.Dense(512))
    model.add(tf.keras.layers.LeakyReLU())
    model.add(tf.keras.layers.Dense(256))
    model.add(tf.keras.layers.LeakyReLU())
    model.add(tf.keras.layers.Dense(1, activation='sigmoid'))
    return model

discriminator = make_discriminator_model()

# Compile the discriminator
discriminator.compile(loss='binary_crossentropy',
optimizer=tf.keras.optimizers.Adam(0.0002, 0.5), metrics=['accuracy'])

# Freeze the weights of the discriminator
discriminator.trainable = False
# Define the combined model for training the generator
def make_gan(discriminator, generator):
    model = tf.keras.Sequential()
    model.add(generator)
    model.add(discriminator)
    return model

gan = make_gan(discriminator, generator)

# Compile the combined model
gan.compile(loss='binary_crossentropy',
optimizer=tf.keras.optimizers.Adam(0.0002, 0.5))
# Train the GAN
def train(gan, discriminator, generator, train_data, batch_size=128,
epochs=100):
    half_batch = int(batch_size / 2)
    for epoch in range(epochs):
```

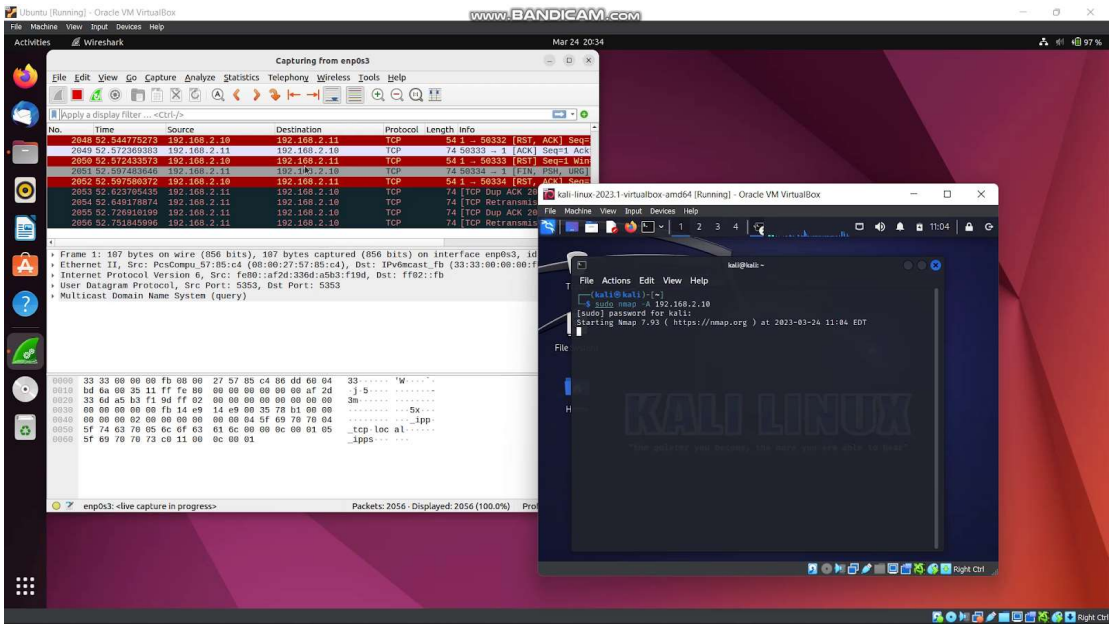
```
        # Train the discriminator
        idx = np.random.randint(0, train_data.shape[0], half_batch)
        real_data = train_data[idx]
        noise = np.random.normal(0, 1, (half_batch, 100))
        fake_data = generator.predict(noise)
        real_labels = np.ones((half_batch, 1))
        fake_labels = np.zeros((half_batch, 1))
        d_loss_real = discriminator.train_on_batch(real_data, real_labels)
        d_loss_fake = discriminator.train_on_batch(fake_data, fake_labels)
        d_loss = 0.5 * np.add(d_loss_real, d_loss_fake)

        # Train the generator
        noise = np.random.normal(0, 1, (batch_size, 100))
        fake_labels = np.ones((batch_size, 1))
        g_loss = gan.train_on_batch(noise, fake_labels)

        # Print progress
        print("Epoch: %d [D loss: %f, acc.: %.2f%%] [G loss: %f]" % (epoch +
1, d_loss[0], 100 * d_loss[1], g_loss))

train(gan, discriminator, generator, train_data, epochs=10000)
# Generate new data
noise = np.random.normal(0, 1, (100, 100))
generated_data = generator.predict(noise)
# Plot the generated data
plt.scatter(generated_data[:, 0], generated_data[:, 1],
c=np.argmax(generated_data[:, 2:], axis=1))
plt.show()
# Save the discriminator model
discriminator.compile(loss='binary_crossentropy',
optimizer=tf.keras.optimizers.Adam(0.0002, 0.5))
discriminator.save('../ML/GAN/discriminator29.h5')
# Save the generator model
generator.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
generator.save('../ML/GAN/generator29.h5')

# Save the combined model
gan.compile(loss='binary_crossentropy',
optimizer=tf.keras.optimizers.Adam(0.0002, 0.5))
```

```
gan.save('../ML/GAN/gan29.h5')
# Load the discriminator model
loaded_discriminator =
tf.keras.models.load_model('../ML/GAN/discriminator29.h5')
# Load the generator model
loaded_generator =
tf.keras.models.load_model('../ML/GAN/generator29.h5')
# Load the combined model
loaded_gan = tf.keras.models.load_model('../ML/GAN/gan29.h5')
# Tensorflow Warning Fix
@tf.function(input_signature=(tf.TensorSpec(shape=[None],
dtype=tf.int32),))
def next_collatz(x):
    print("Tracing with", x)
    return tf.where(x % 2 == 0, x // 2, 3 * x + 1)
honeypot_result = pd.read_csv("../honey.csv")
# Preprocess the data
def result_preprocess(honey_res):
    categorical_columns = [col for col in honey_res.columns if
honey_res[col].dtype == 'object']
    for col in categorical_columns:
        honey_res[col] = honey_res[col].astype('category').cat.codes
    honey_res.to_csv("./realtime.csv")
    return honey_res.values

honey_res = result_preprocess(honeypot_result)
```

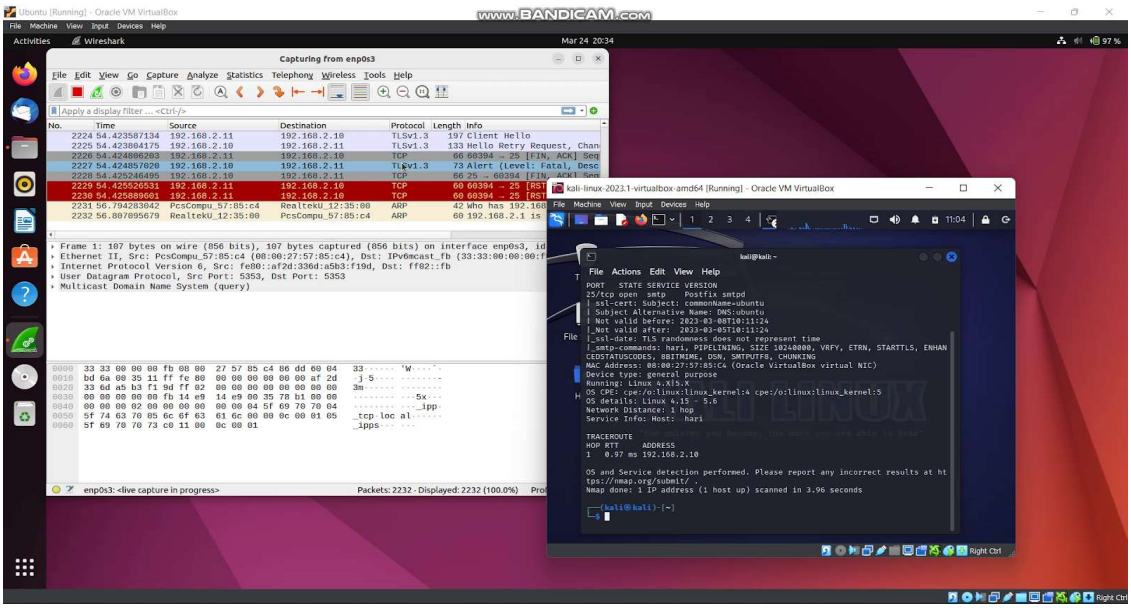
7.2 SNAPSHOTS



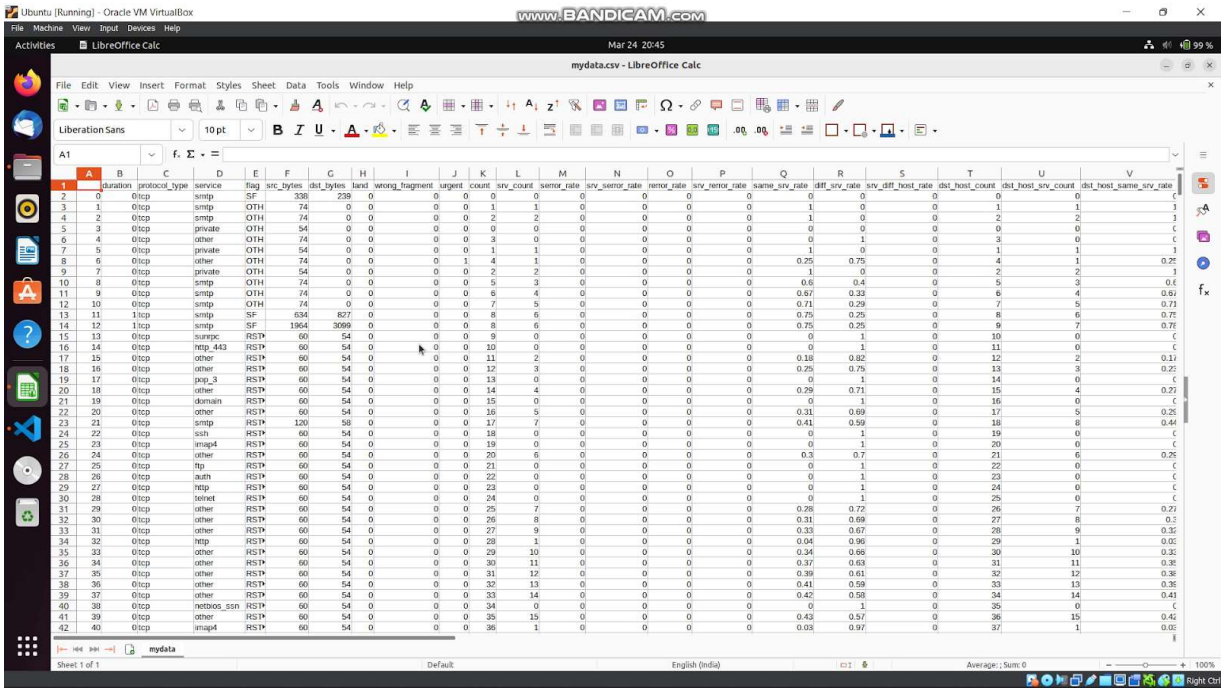
Fig, 7.2.1 Before attack



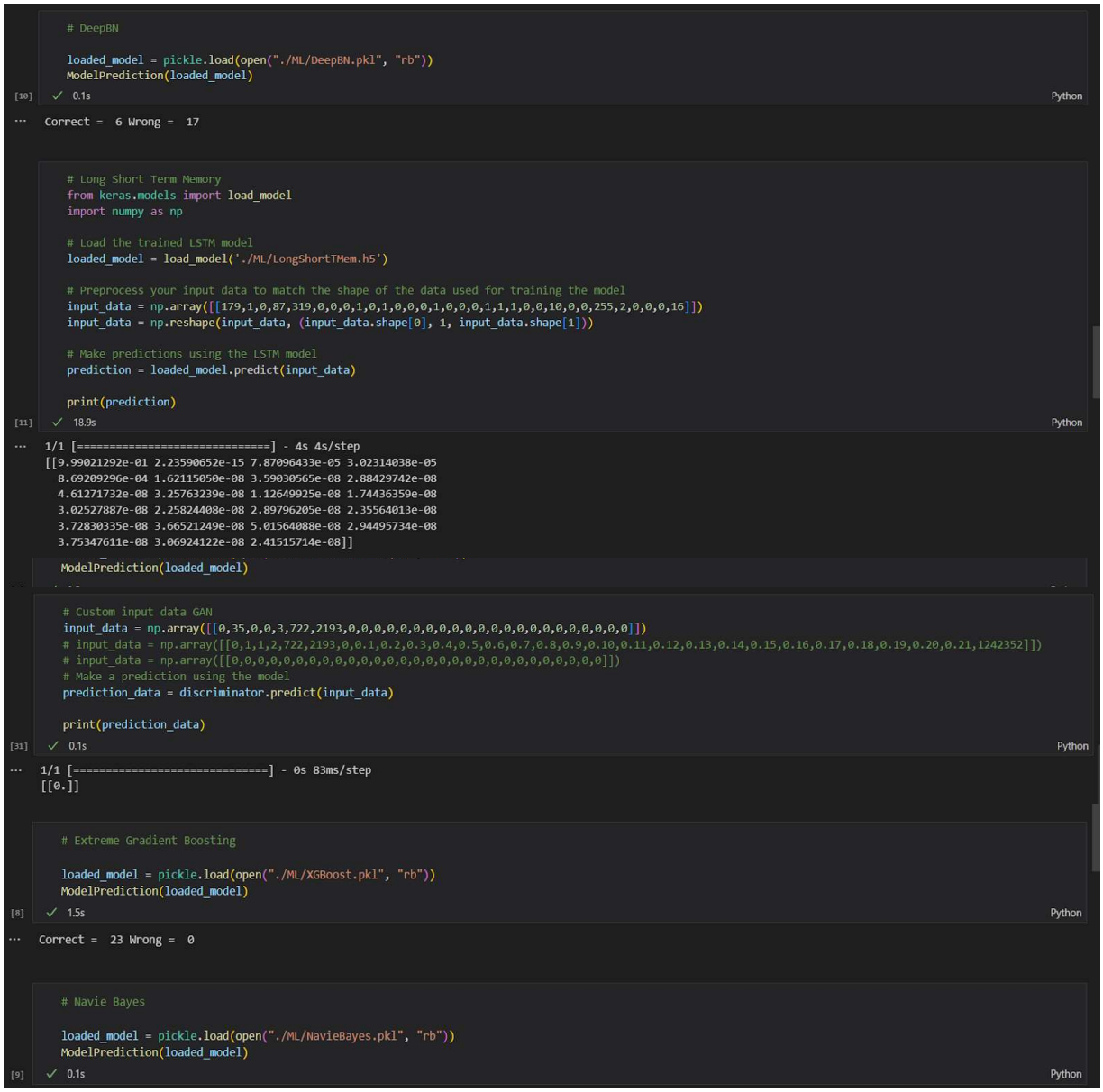
Fig, 7.2.2 Attacking



Fig, 7.2.3 After attack



Fig, 7.2.4 Feature Extraction



Fig, 7.2.5 ML Prediction Result

CHAPTER 8
REFERENCES

[1] Mr. Mohit Tiwari, Raj Kumar, Akash Bharti, Jai Kishan (2017) Intrusion Detection System, International Journal of Technical Research and Applications e-ISSN: 2320-8163.

[2] Gozde Karatas, Onder Demir, Ozgur Koray Sahingoz, "Deep Learning in Intrusion Detection Systems", *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, pp.113-116, 2018.

[3] Nilesh Kunhare, Ritu Tiwari, "Study of the Attributes using Four Class Labels on KDD99 and NSL-KDD Datasets with Machine Learning Techniques", *2018 8th International Conference on Communication Systems and Network Technologies (CSNT)*, pp.127-131, 2018.

[4] Basant Subba, "A Neural Network based NIDS framework for intrusion detection in contemporary network traffic", *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp.1-6, 2019.

[5] Mridula Sharma, Haytham Elmiligi, Fayez Gebali, Abhishek Verma, "Simulating Attacks for RPL and Generating Multi-class Dataset for Supervised Machine Learning", *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp.0020-0026, 2019.

[6] Sun N, Zhang J, Rimba P, Gao S, Zhang LY, Xiang Y, Data-driven cybersecurity incident prediction: A survey, IEEE Commun. Surv. Tutor. 21 (2) 1744–1772 (2019)

[7] Akshat Divya, Anchit Bhushan, Nihal Anand, Rishabh Khemka, Sumithra Devi K.A.H (2020) HONEYPOT: Intrusion Detection System. International Journal of Education, Science, Technology, Engineering vol. 3, no. 1, pp. 13-18, June 2020.

[8] Husak M, Bartos V, Sokol P, Gajdos A, Predictive methods in cyber defense: Current experience and research 43 challenges, Future Generation Computer Systems, Volume 115, 517-530 (2021)

[9] Philip Wester, Fredrik Heiding, Robert Lagerström, "Anomaly-based Intrusion Detection using Tree Augmented Naive Bayes", *2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW)*, pp.112-121, 2021.

[10] Ritu Bala, Ritu Nagpal, "Intrusion Detection Based on Decision Tree Using Key Attributes of Network Traffic", *Applications of Artificial Intelligence and Machine Learning*, vol.778, pp.583, 2021.

[11] Bo-Xiang Wang, Jiann-Liang Chen (2022): An AI-Powered Network Threat Detection System Doi 10.1109/ACCESS.2022.3175886

[12] Ruizhe Zhao, Yingxue Mu, Long Zou, Xiumei Wen, "A Hybrid Intrusion Detection System Based on Feature Selection and Weighted Stacking Classifier", *IEEE Access*, vol.10, pp.71414-71426, 2022.

[13] Suzanne Widup, Alex Pinto, Gabriel Bassett, David Hylender (2021) Verizon Data Breach Investigations Report, May 2021.

[14] WidupSuzanne, WidupMarc, SpitlerDavid Hylender Gabriel (2018). “Verizon Data Breach Investigations Report 2022”. April 2022.

Honeypot-based Cyber Attack Detection

Mrs. M Karthigha[1], Abishek PS[2], Hari Kishore VP[3], Kugaanesen S[4]
[1-4] Sri Ramakrishna Engineering College / Department of Computer Science and Engineering, Coimbatore, India
[1-4] Email: {karthiga.m@srec.ac.in, abishek292001@gmail.com, hari.1901043@srec.ac.in, kugaanesen.1901068@srec.ac.in}

***Abstract* — Cyber Intrusion is the most threatening expression in the cyber world, and it is a dangerous crime that many corporations and individuals who are a part of the Cyber World are horrified of. It results from not only a financial loss but also includes personal data which is impacted as a flooded river when the data is exposed in a data breach. Cyber Intrusion Detection System is a technology used to identify and alert unauthorised access to a network system or a network device. It analyses network traffic and logs files maintained by a device and reports or alerts when an outsider is trying to gain access to a network. Honeypot is a technology that acts as a catchy pot of honey for an attacker. When an attacker tries to catch the pot, the Honeypot system will alert the administrator and block it. Both technologies can be combined with Machine Learning to automate and improve the prediction rate so that attackers will be prevented. The use of Machine Learning algorithms detects the type of attacks. Further research should be conducted for the results to use the combination of honeypots, Cyber Intrusion, and Machine Learning to detect Cyber Attacks and the advancements, efficiency, and correctness of the prediction.**

***Index Terms*— Honeypot, Machine Learning, Cyber Attack, Data Breach, Attack Prevention.**

I. Introduction

The world is evolving with cutting-edge technology, and people around the world are interconnected with each other through the internet with the help of electronic devices and smart gadgets. There are about 5 billion active internet users worldwide. A major threat to Internet users is cyberattacks. Cyber-attacks may lead to any risk which includes financial losses, personal data leaks, business-related problems, corporate security, and mainly personal data security. Data Breach sometimes lights up a few corporate secret crimes which are more commonly found these days. Though it lights up crimes, Data Breach is a crime that involves one's or some personal details.

A honeypot is a cybersecurity mechanism used to detect, deflect, or counteract unauthorised use of information systems. A honeypot is essentially a decoy system that is designed to attract and trap potential attackers by imitating a vulnerable system or application.

The idea behind a honeypot is to give attackers a fake system to attack instead of the real one, allowing security researchers to observe the attacker's techniques and tactics without risking damage to real systems. When an attacker interacts with a honeypot, the honeypot captures information about the attack, including the attacker's IP address, methods, and tools used in the attack. There are various types of honeypots, including high-interaction and low-interaction honeypots. High-interaction honeypots are designed to provide a realistic environment for attackers to operate in, whereas low-interaction honeypots simulate only certain aspects of a system. Honeypots can be used as a proactive security measure to detect and prevent attacks, as well as a research tool to study and understand attackers' behaviour and motives.

The global average cost of a Data breach is about 4.35 USD. Verizon's 2022 Data Breach Investigation Report (DIBR) states that about 62% of incidents are of System Intrusion patterns involving threat actors compromising partners. 13% increase in Ransomware which is more than the combined past 5 years. Avoiding such attacks for an individual is impossible since targeted attacks are more common these days where the data of high-powered people are not safe. To avoid such attacks in a network, the use of new technologies is a must to get more secure. The attackers are smarter as there are a few ways to bypass the honeypots.

Data can be secured in many ways, but the most efficient in terms of power consumption, data consumption, and quick response is more important. The combination of Honeypot and Cyber Intrusion Detection ways using Machine Learning Algorithms can predict almost every input data frame in less time with a high prediction rate which can stop the attacker from accessing the data.

II. Proposed Methodology

Machine learning approaches with a change in the network setup can be used to predict the Honeypot log data and whether any attacker is trying to gain access or not in an efficient way. It is achieved by using the required data from the honeypot log files. It requires a few features which are extracted from the generated log report from the Honeypot. The main purpose of this methodology is to obtain high accuracy and low processing power.

The prediction accuracy of an output of a Machine Learning algorithm is strongly related to the quality of its dataset. The proposed methodology uses the "KDDCUP99" dataset provided by Fifth International

Conference on Knowledge Discovery and Data Mining. The KDDCUP99 dataset is used with ML models directly from the honeypot logs. The dataset has labels and hence it is a supervised model, we test its accuracy with all suitable Machine Learning Algorithms.

A. Network Design

The network design is designed in such a way that all the data is the hidden backside of a honeypot server divided by firewalls. Multiple layered designs are required to attract more attackers and it helps in identifying attackers which use honeypot bypass techniques. It is explained well in the following figure with network design designed using Cisco Packet Tracer.

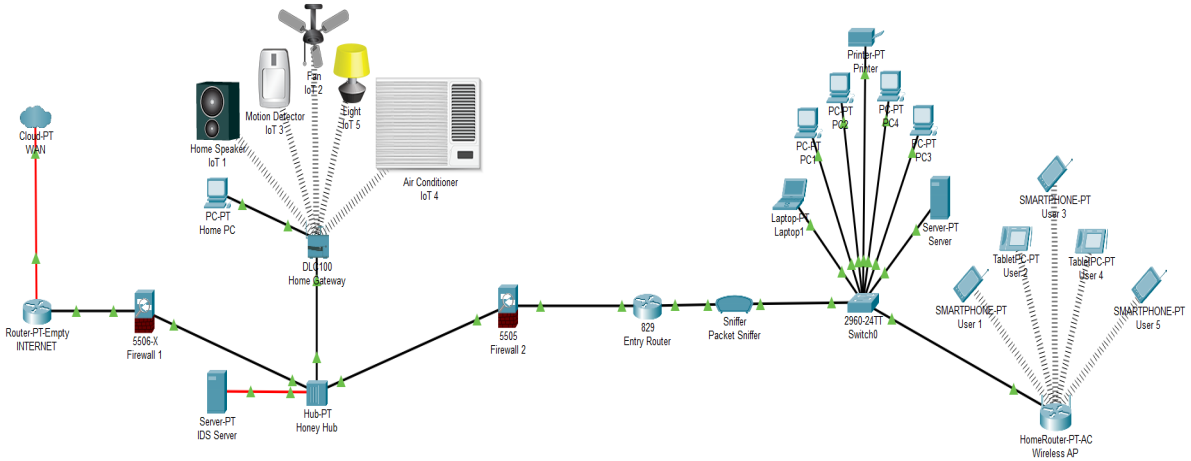


Fig. 2.1 Network Design

i. The whole network is entered directly through the Internet router from the Internet Service Provider (ISP) or the outer network.

ii. All the traffic from the Internet Service Provider enters the Internet Router and the data packets are sent over Firewall 1 to filter the usual unwanted packets from the internet or the outer network.

iii. Filtered packets then enter the Honey Hub. The honey Hub is connected to a home gateway that is fully connected with more devices using the internet (IoT Devices), this may be physical devices or may be virtual devices running old or vulnerable operating systems. Old and vulnerable software versions for IoT devices are the 1st layer of trap for attackers. Since the software is vulnerable by nature, they are more likely to act as a natural honeypot.

iv. The role of the Honey HUB is to transmit all packets to the IDS server since HUB broadcasts every packet detail to every device connected to it.

v. The honeypot server which is connected via optical fibre receives data more quickly since they are much faster than the normal twisted pair of Cat 5, Cat 6, Cat 7, and Cat7E cables.

vi. IDS server analyses the network and uses Machine Learning Algorithm that predicts the packet's nature and collects all required details for the prediction and sends the report to Firewall 2 via Honey Hub with an encrypted secured channel.

vii. Entry Router receives packets that are filtered by Firewall 2 as directed by the honeypot server. The packets move via a packet sniffer which collects all data for future study purposes since “Nothing in the internet is 100% safe”.

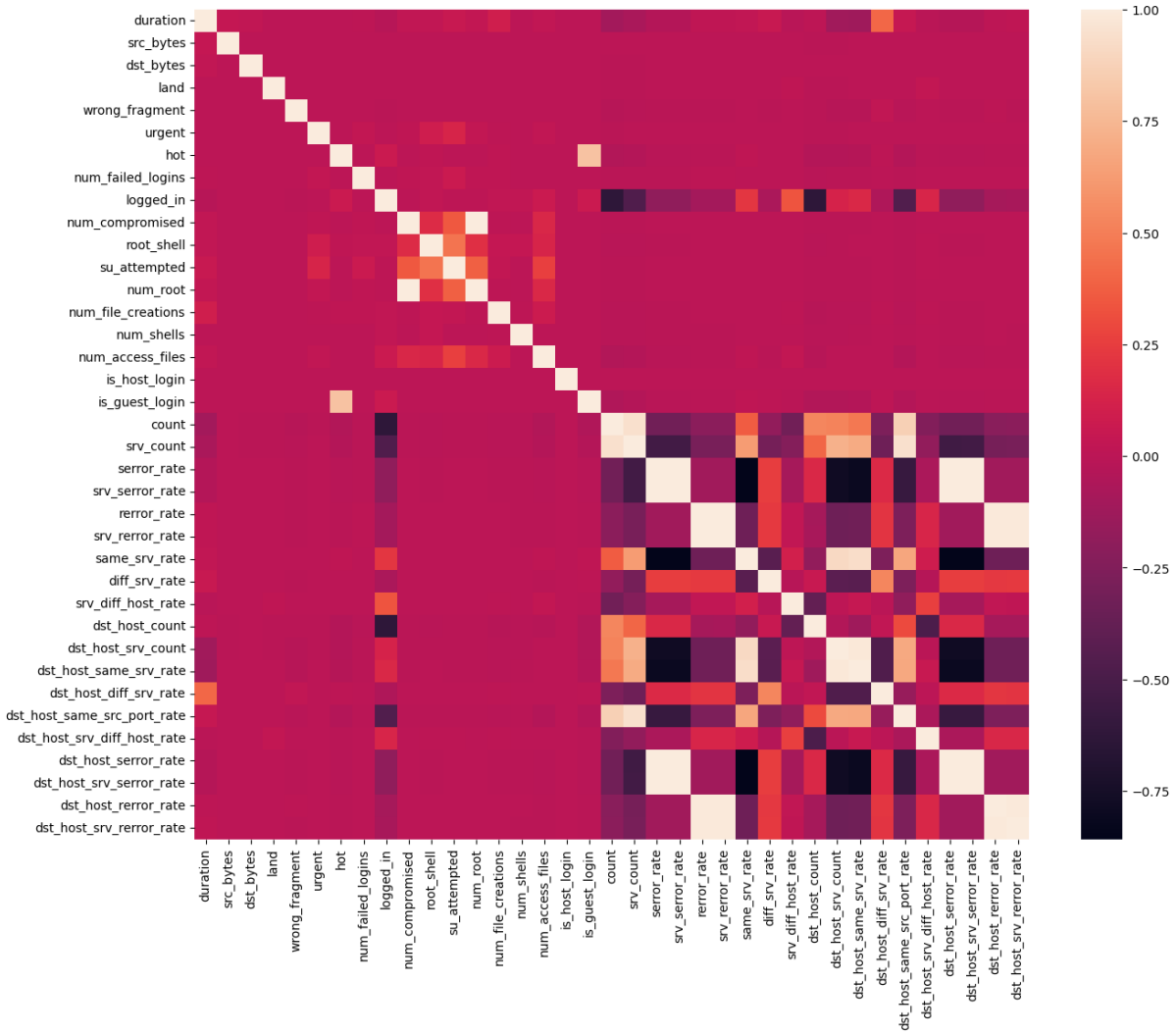
viii. Finally, the Switch0 and the Home Router or the regular network are connected and can be used normally.

B. Dataset

The dataset used is the one released by The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining. The task was to build a network intrusion detector, a predictive model capable of distinguishing between “bad"” connections, called intrusions or attacks, and “good” normal connections. The dataset has 42 columns which are the main features that determine the prediction. The most important field mainly consists of attack type and type of connection are as follows:

- ‘Normal’ for regular data.
- ‘dos’ type of attack for ‘Neptune, smurf, land, back, pod, teardrop’.
- ‘probe’ type of attack for the type ‘ipsweep, satan, port sweep or nmap’.
- ‘r2l’ type of attack for type ‘ftp_write, guess_passwd, imap, phf, spy, warezclient, warezmaster or multihop’,
- ‘u2r’ for ‘Perl, loadmodule, buffer_overflow or rootkit’

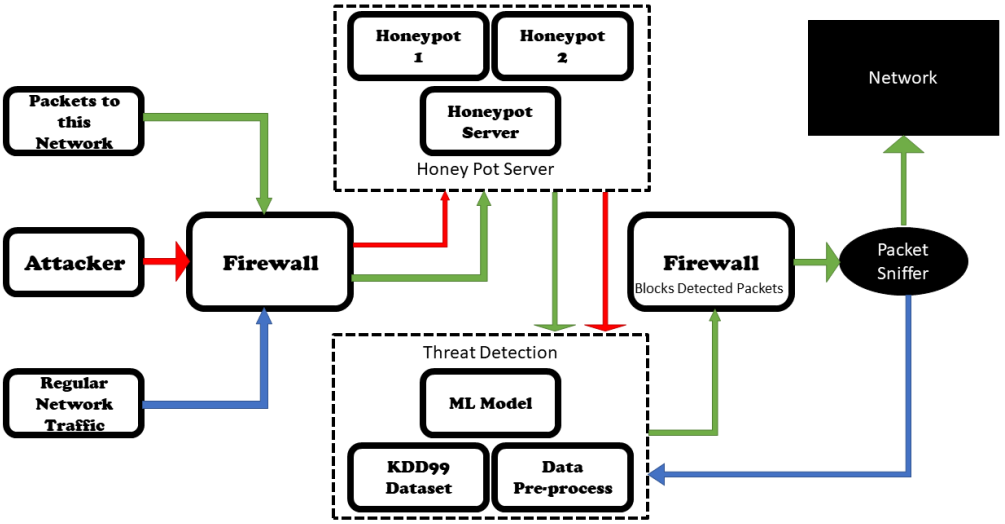
The database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, but the combination of honeypots with a 2 layered intrusion model using different ML Algorithms helps in identifying more efficiently and with low processing power. The following figure 2.2 shows the heatmap of all the features of the KDD99 dataset for all 42 features.



Fig, 2.2 KDD99 Heatmap

C. Intrusion Detection System by Machine Learning

The common packets which are filtered by Firewall are almost safe from regular intrusions and regular web traffic. The firewall may be a hardware firewall or else a software-based firewall to filter the packets. All the data together come to the HUB and are broadcasted to every device. A smart attacker can sense the use or any other mode of packet sniffing, and even the intruder can modify or destroy log files and reports generated by the Honeypot Server, but the use of Hub makes it look like a natural device used for connecting devices. The possibility of identifying IoT devices is very high because of the vulnerable software running in them. All the data is monitored by software or hardware in the Honeypot server. The figure 2.3 explains well as a block diagram.



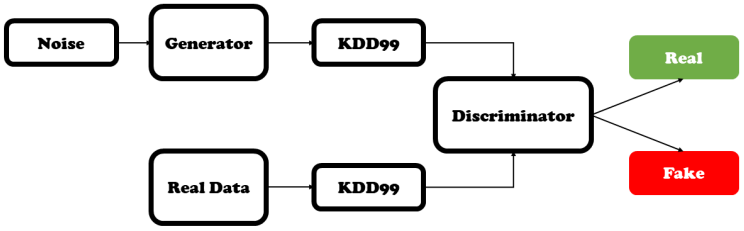
Fig, 2.3 Block diagram

Honeypot Server is the main part that determines whether the packets are safe or not. It analyses every single packet in detail and gives predictions using Machine Learning Algorithm. Every machine learning model has its specific type of processing method. Every classifier has its working algorithms. As the desired output is a prediction of type "intruder" or "Normal" type, a proper machine learning model is to be selected for the required output. A set of ML classifiers are taken to test the dataset. The dataset has a label column which is suitable for supervised machine-learning algorithms. The machine learning algorithms used are Gaussian Naive Bayes, Decision Tree, Random Forest, Gradient Boosting, Extreme Gradient Boosting, Generative adversarial network, Linear Regression, Logistic Regression, Long Short-Term Memory, and Deep Belief Network.

D. Threat Detection

Different Machine Learning algorithms are used to determine whether the connection is good or it is an intruder. The data from the honeypot is analysed well by the Machine Learning model which has about 42 features used for prediction. There are tested using various Machine Learning models such as Gaussian Naive Bayes, Decision Tree, Random Forest, Gradient Boosting, Extreme Gradient Boosting, Generative adversarial network, Linear Regression, Logistic Regression, Long Short-Term Memory, and Deep Belief Network as mentioned above. The one which is suitable for intrusion detection gives a high accuracy value and the high accuracy model is selected for the detection.

Generative Adversarial Networks (GANs) are a class of deep learning algorithms used to generate new and synthetic data by pitting two neural networks against each other in a zero-sum game framework. There are two main components: a generator and a discriminator. The generator takes a random noise vector as input and maps it to an output that is meant to resemble the target data distribution. The discriminator takes in both real samples from the target data distribution and fake samples generated by the generator and tries to distinguish between the two, they are trained in an adversarial manner, generator creating samples that are realistic to the discriminator, the discriminator tries to classify the samples as either real or fake. Over time, the generator improves its ability to create realistic samples, and the discriminator improves its ability to identify fake samples.



Fig, 2.4 GAN Block Diagram

Since all the classifiers perform well for the used dataset, we need to consider some other factors as the prediction is used for security purposes and threat detection. The dataset has 48,98,430 columns and 42 Rows with different features. As the predictions are used in security and threat detection applications, we need to take care more of the input dataset. Generative Adversarial Networks can be used if their accuracy is good enough compared to other classifiers.

III. Results

Realtime Honeypots capture intruders by various methods and the data required for the prediction is generated by it. As a whole, it is passed to the various Machine Learning models to get its performance and accuracy. Test and train time is also considerable for power-efficient working.

The Decision Tree algorithm shows a test accuracy of 99.62% and a training time of 10.85s. The Linear Regression algorithm shows a test accuracy of 97.75% and a training time of 5.99s. The Gaussian Naive Bayes algorithm shows a test accuracy of 93.21% and a training time of 3.74s. The Logistic Regression algorithm shows a test accuracy of 1.41% and a training time of 376.92s. The Random Forest algorithm shows a test accuracy of 99.92% and a training time of 495.71s. The Gradient Boosting algorithm shows a test accuracy of 95.94% and a training time of 6372.18s. The Extreme Gradient Boosting algorithm shows a test accuracy of 98.93% and a training time of 443.02s. The Long Short-Term Memory algorithm shows a test accuracy of 97.87% and a training time of 7966.02s. The Deep Belief Networks algorithm shows a test accuracy of 79.84% and a training time of 819.10s. The Generative Adversarial Networks algorithm shows a test accuracy of 99.99% and a training time of 130.02s. The table 3.1 shows the results of all the classifiers and its training time with testing accuracy of each.

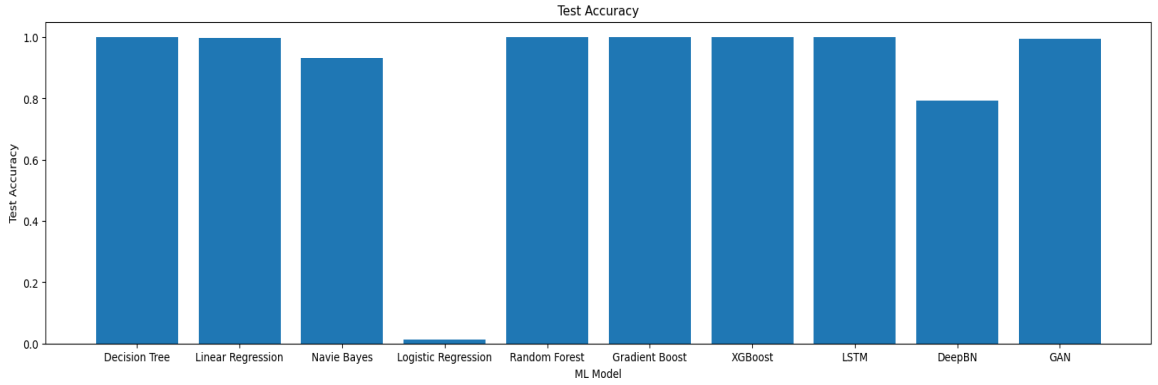
Classifier	Test Accuracy	Training Time
Decision Tree	0.9962	10.846
Linear Regression	0.9775	5.988
Navie Bayes	0.9321	3.739
Logistic Regression	0.0141	376.921
Random Forest	0.9992	495.712
Gradient Boost	0.9594	6372.185
XG Boost	0.9893	443.016
LSTM	0.9787	7966.016
Deep BN	0.7984	819.102
GAN	0.9999	130.016

Table, 3.1 Classifiers Performance

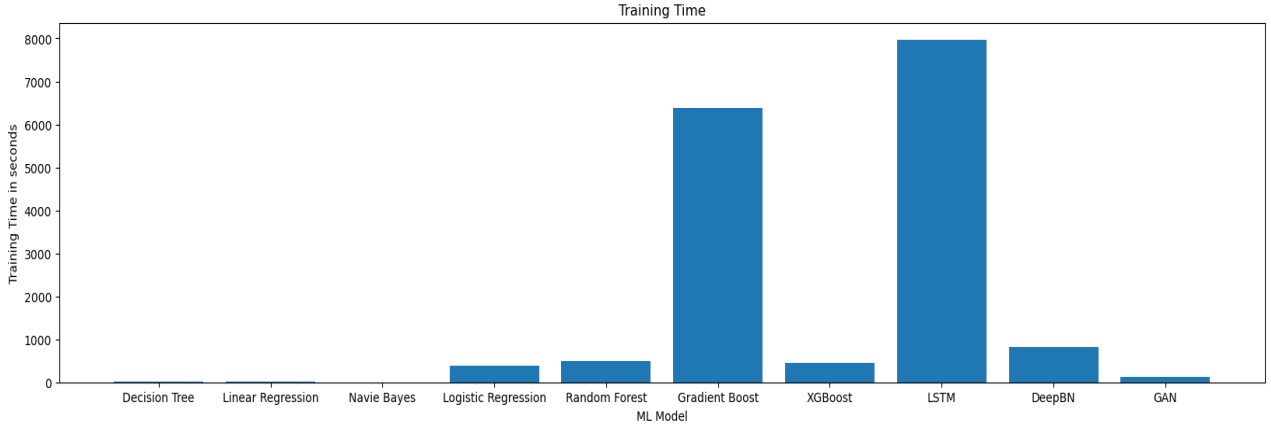
Epoch	Discriminator Loss	Generator Loss	Accuracy
1	38.05141	0.689847	42.19
100	0.131214	1.496475	100
200	0.096201	1.761345	100
300	0.025205	3.041839	100
400	0.011418	3.800014	100
500	0.005777	4.490452	100
600	0.00349	4.992599	100
700	0.002289	5.401649	100
800	0.001711	5.725489	100
900	0.001114	6.107335	100
1000	0.000896	6.343067	100

Table, 3.2 GAN Results

The graph 3.1 and 3.2 specifies the testing accuracy and the training time of the different classifiers as bar graph. In testing accuracy, Logistic Regression has performed the least and almost all other classifiers score more than 80. The bar graph says more about the training time required by all the classifiers.



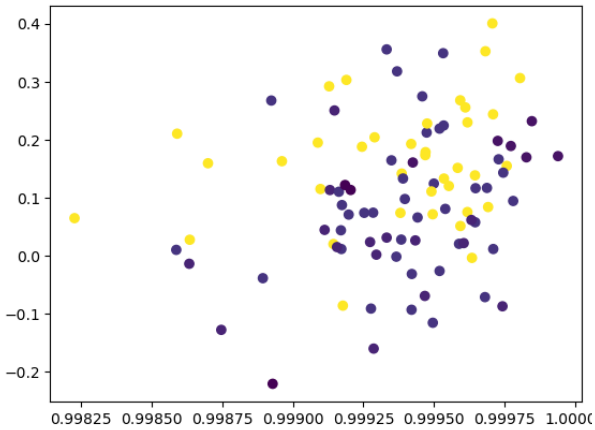
Fig, 3.1 Testing Accuracy



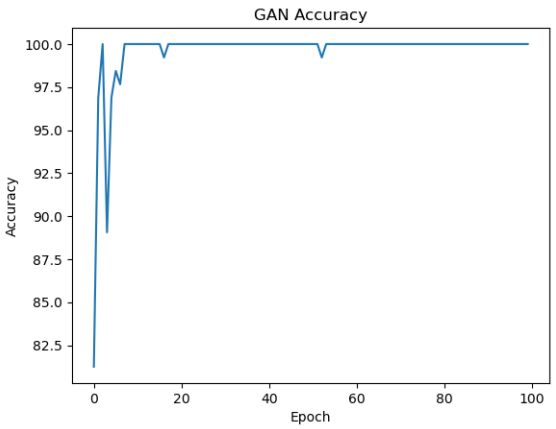
Fig, 3.2 Training Time (in seconds)

IV. Conclusions

The use of Generative Adversarial Networks is used for intrusion detection. The maximum-performing algorithm exists but it is limited by the size of the dataset. Since Generative Adversarial Networks are trained using synthetically generated values, their prediction and accuracy are still high. Other models have been trained with just the input dataset of size 48,98,430 columns. The GAN gives accurate results, compared to all other algorithms GAN is trained with a high number of inputs with about 1000 epochs. The use of network design is to reduce the regular traffic and to catch the attackers. Future research should verify the new methods of attackers and include more features for advanced attacks.



Fig, 4.1 GAN Generated Data



Fig, 4.2 GAN Accuracy

The Generative Adversarial Network gives the most accurate result for the KDD99 dataset with almost more accuracy that, most of the accuracy plot in the graph falls within 1 and 0.99825 as shown in the figure 4.1

and figure 4.2. Since synthetic data is used to train it, it is more accurate than any classifiers used for the same dataset. Thus, Generative adversarial network has been selected for the Intrusion Detection System.

References

[1] Mr. Mohit Tiwari, Raj Kumar, Akash Bharti, Jai Kishan (2017) Intrusion Detection System, International Journal of Technical Research and Applications e-ISSN: 2320-8163.

[2] Gozde Karatas, Onder Demir, Ozgur Koray Sahingoz, "Deep Learning in Intrusion Detection Systems", *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, pp.113-116, 2018.

[3] Nilesh Kunhare, Ritu Tiwari, "Study of the Attributes using Four Class Labels on KDD99 and NSL-KDD Datasets with Machine Learning Techniques", *2018 8th International Conference on Communication Systems and Network Technologies (CSNT)*, pp.127-131, 2018.

[4] Basant Subba, "A Neural Network based NIDS framework for intrusion detection in contemporary network traffic", *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp.1-6, 2019.

[5] Mridula Sharma, Haytham Elmiligi, Fayez Gebali, Abhishek Verma, "Simulating Attacks for RPL and Generating Multi-class Dataset for Supervised Machine Learning", *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp.0020-0026, 2019.

[6] Sun N, Zhang J, Rimba P, Gao S, Zhang LY, Xiang Y, Data-driven cybersecurity incident prediction: A survey, IEEE Commun. Surv. Tutor. 21 (2) 1744–1772 (2019)

[7] Akshat Divya, Anchit Bhushan, Nihal Anand, Rishabh Khemka, Sumithra Devi K.A.H (2020) HONEYPOT: Intrusion Detection System. International Journal of Education, Science, Technology, Engineering vol. 3, no. 1, pp. 13-18, June 2020.

[8] Husak M, Bartos V, Sokol P, Gajdos A, Predictive methods in cyber defense: Current experience and research 43 challenges, Future Generation Computer Systems, Volume 115, 517-530 (2021)

[9] Philip Wester, Fredrik Heiding, Robert Lagerström, "Anomaly-based Intrusion Detection using Tree Augmented Naive Bayes", *2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW)*, pp.112-121, 2021.

[10] Ritu Bala, Ritu Nagpal, "Intrusion Detection Based on Decision Tree Using Key Attributes of Network Traffic", *Applications of Artificial Intelligence and Machine Learning*, vol.778, pp.583, 2021.

[11] Aakash Singh, Parth Kitawat, Shubham Kejriwal, Swapnali Kurhade, "Intrusion Detection System Using Homomorphic Encryption", *Intelligent Data Communication Technologies and Internet of Things*, vol.101, pp.505, 2022.

[12] Bo-Xiang Wang, Jiann-Liang Chen (2022): An AI-Powered Network Threat Detection System Doi 10.1109/ACCESS.2022.3175886

[13] Ruizhe Zhao, Yingxue Mu, Long Zou, Xiumei Wen, "A Hybrid Intrusion Detection System Based on Feature Selection and Weighted Stacking Classifier", *IEEE Access*, vol.10, pp.71414-71426, 2022.

[14] Zebin, Tahmina; Rezvy, Shahadate; Luo, Yuan (2022): An Explainable AI-based Intrusion Detection System for DNS over HTTPS (DoH) Attacks. TechRxiv. Preprint.