Honeypot-based Cyber Attack Detection

Mrs. M Karthigha1, Abishek PS2, Hari Kishore VP3, Kugaanesen S4

1-4 Sri Ramakrishna Engineering College / Department of Computer Science and Engineering, Coimbatore, India

1-4 Email: {karthiga.m@srec.ac.in, abishek.1901002@srec.ac.in, 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 refers to a technology utilized 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.**

***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 [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 unauthorized 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.

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. Related Literature

Honeypots have been a topic of research for several years and have seen significant advancements in recent times, particularly with the integration of artificial intelligence (AI) techniques for enhanced detection and prevention capabilities. In this literature survey, we will explore various honeypot research papers and AI-based honeypot research papers to understand the current state of honeypot technology and how AI can be leveraged to improve honeypot capabilities. Experts have done a lot of work in this field. [7] provides a detailed examination of honeypots as a security mechanism for network-based attack prevention and detection. The authors describe the advantages, implementation, and use of honeypots as an effective information-gathering tool for network attacks. The study includes methodologies like roaming honeypots and backpropagation algorithms to identify and prevent source-address parodying DDoS attacks. In [1] the author discusses the importance of Intrusion Detection System (IDS), which helps monitor network and system activities and detects malicious activities. The paper explores various types of IDS, such as network-based and host-based IDS, along with their components and functioning. The IDS tool can detect and prevent intrusion from attackers, making it a crucial security system for organizations in today's internet-driven world. [4] gives a Neural Network based NIDS framework for real-time anomaly detection in modern-day network traffic, using the UNSW-NB15 dataset for training. The framework uses convex Logistic Regression cost functions, along with stochastic gradient descent and simulated annealing to fine-tune hyperparameters of the classifier. [10] discusses the use of data mining techniques in implementing intrusion detection systems (IDS) to protect computer networks from attacks. The study compares the performance of two decision tree algorithms, J48 and Random Forest, using attribute selection methods Information Gain and Rankers Algorithm

Ⅲ. Our Contribution

The following are some of the key contributions and findings of our work:

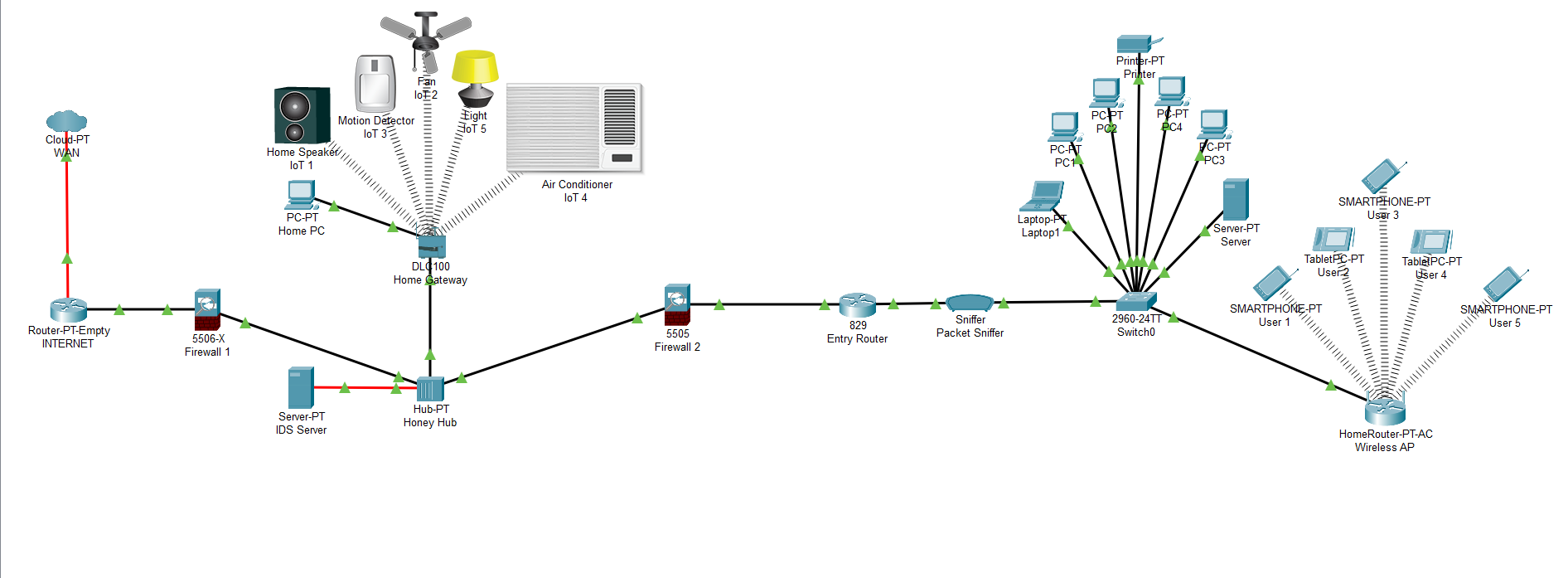
1. Firstly, a modelled network design is created for trapping intruders naturally. A Honeypot can be found by some intelligent intruder. A network design which mimics like malicious to attackers but has running honeypot in it creates multiple levels of security hidden in the network.
2. The whole network is monitored using a network monitoring tool and the features are separated for machine learning.
3. We proposed a deep learning architecture Generative Adversarial Network (GAN) consisting of two neural networks: a generator and a discriminator. The generator network learns to generate new data that resembles the original data, while the discriminator network learns to distinguish between the generated data and the real data.
4. The best results are identified be running tests with different machine learning algorithms and selected the best performing model.

Ⅳ. Proposed Methodology

Machine learning approach with a unique network setup gives accurate results 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 [7]. The primary objective of this methodology is to achieve high accuracy while utilizing minimal processing power. A Machine Learning algorithm's predictive accuracy is highly dependent on the dataset's quality.

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. 2.1 Network Design*

1. The network design used is Hybrid architecture. We use it to customize the security features.
2. The whole network is entered directly through the Internet router from the Internet Service Provider (ISP) or the outer network.
3. 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.
4. 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.
5. 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.
6. The honeypot server, connected via optical fibre, receives data faster compared to standard twisted pair cables, such as Cat 5, Cat 6, Cat 7, and Cat 7E, due to the former's higher speed capability.
7. 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.
8. 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”.
9. Finally, the Switch0 and the Home Router or the regular network is connected and can be used normally.
10. HONEYPOT

The network design consists of two layers of security mechanism as explained well in the network design. It is important to have multiple layers of security features to ensure more security. Hackers are always more intelligent where they get access if even there is a minute defect in the code or even network design.

* 1. The first layer of security has IoT devices with older software. Older software is more vulnerable than the latest one. IoT devices are more likely to be trapped by hackers to get into a network. If any hacker traps an IoT device, those IP will be blocked by a firewall.
  2. As the honeypot used is a high interaction honeypot, few ports were made open which mimics the open ports of software running inside it such as Microsoft update plugin’s port or database’s default port or TCP, UDP, TELNET, POP3, FTP, SMTP, etc.

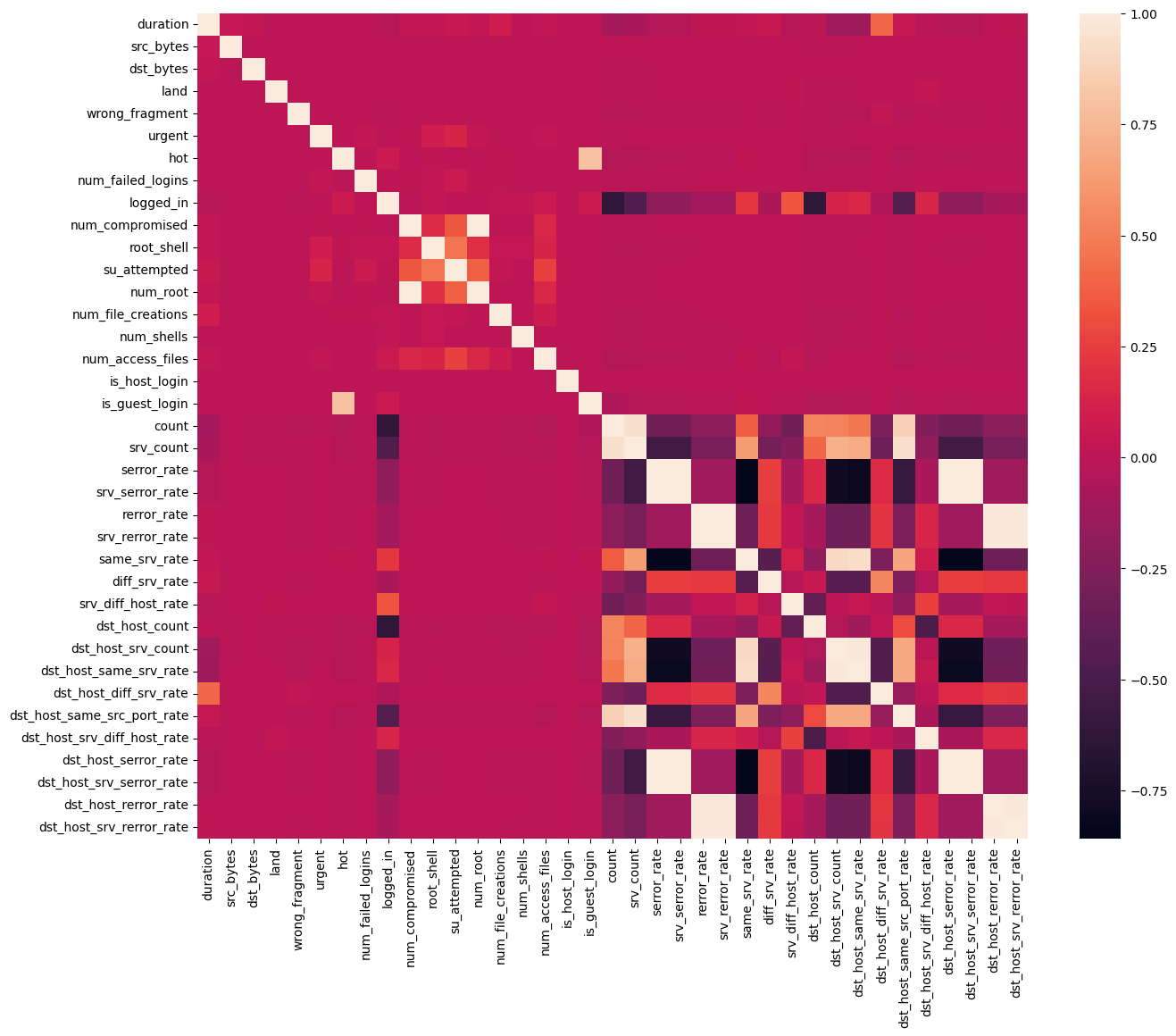
The whole network is monitored and the monitored data is saved. The data is then used for the feature extraction process for the machine learning purposes. The data is saved using network monitoring tools. The network sniffer or packet sniffers traps the whole data where we can find anything there.

1. Dataset

The dataset utilized for this study was released during the Third International Knowledge Discovery and Data Mining Tools Competition, which was held in tandem with KDD-99[3] The Fifth International Conference on Knowledge Discovery and Data Mining. Participants were tasked with constructing a network intrusion detector that could differentiate between "bad" connections (intrusions or attacks) and "good" normal connections by creating a predictive model. 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 [3]:

* ‘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’

Within the database, there exists a standard set of data to undergo auditing, comprising an array of intrusions emulated within 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. Figure 2.2 depicts the heatmap for all 42 features of the KDD99 dataset.

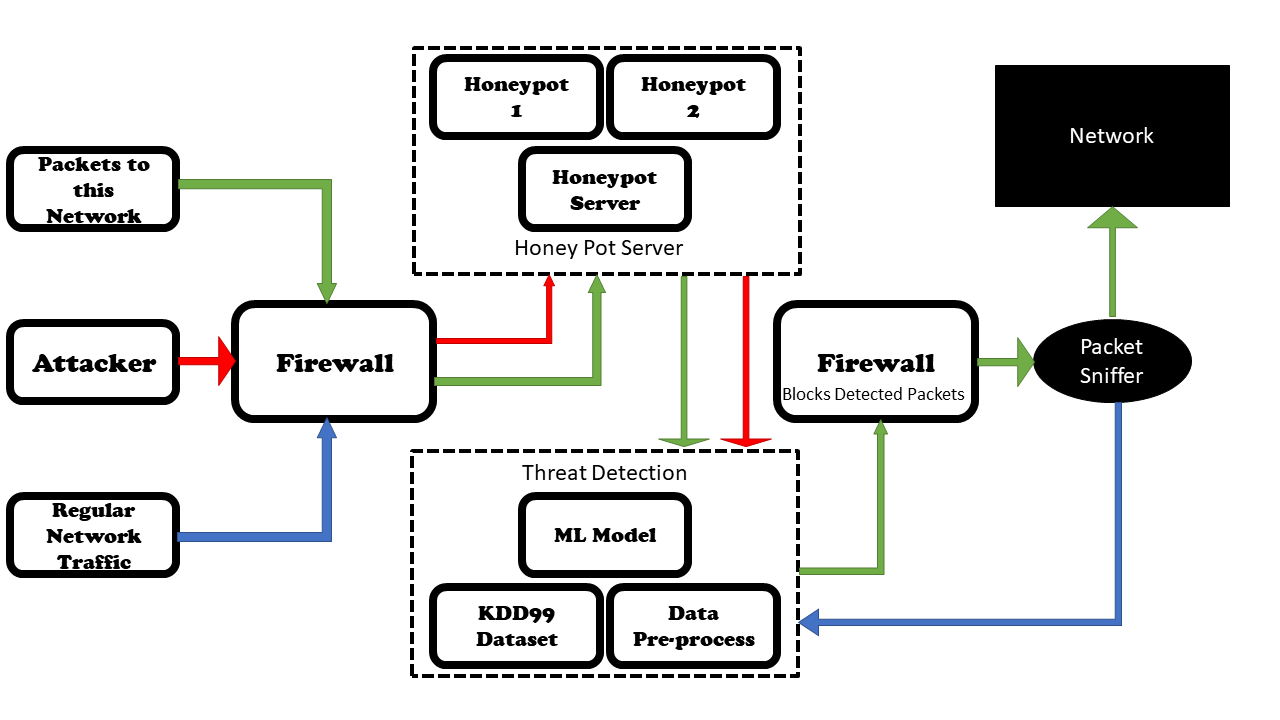


*Fig, 2.2 KDD99 Heatmap*

1. 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.

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 [12]. 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 is having a label column which is suitable for supervised machine-learning algorithms. The utilized machine learning algorithms encompass 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.

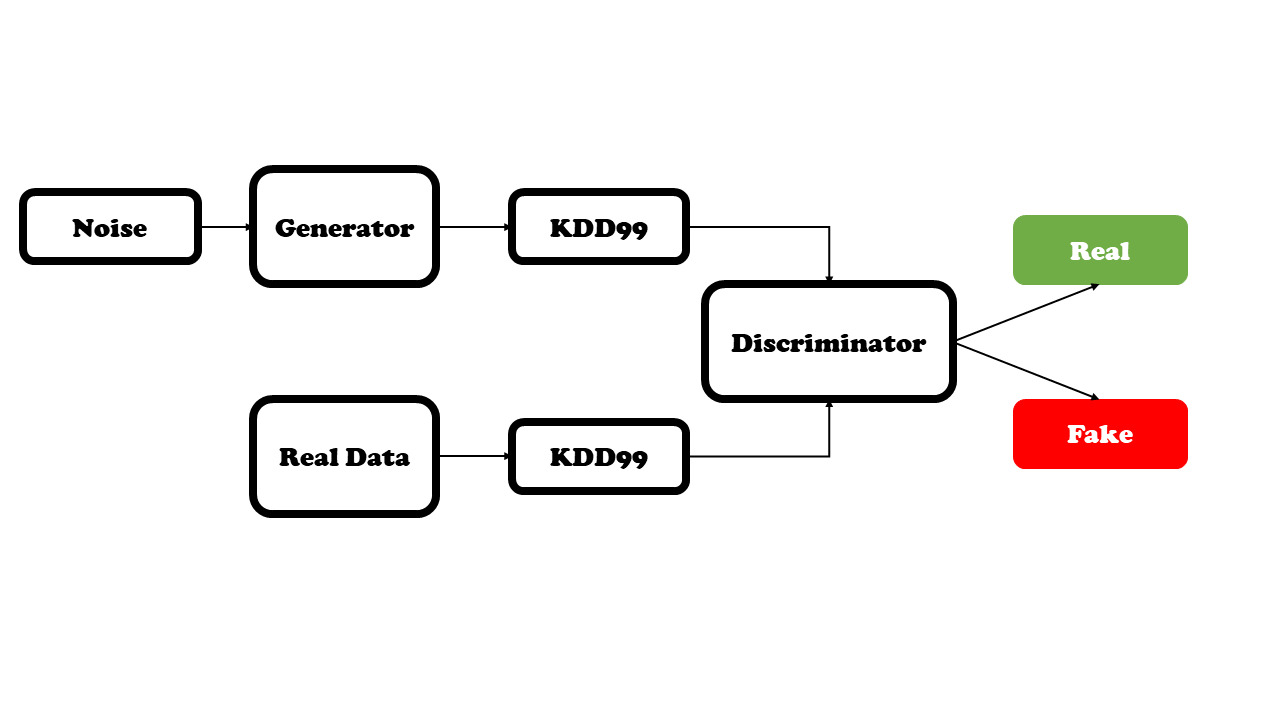


*Fig, 2.3 Block diagram*

1. Threat Detection

Various machine learning algorithms are utilized 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 [12]. There are tested using Different machine learning models like Gaussian Naive Bayes [9], Decision Tree [10], 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.

The application of Generative Adversarial Networks (GANs), a class of deep learning algorithms, involves generating new and synthetic data by pitting two neural networks Within a zero-sum game framework, the entities compete against each other. There are two main components: a generator and a discriminator. Taking a random noise vector as input, the generator maps it to an output that mimics the target data distribution, whereas 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, In an attempt to classify samples as real or fake, the discriminator scrutinizes them, while the generator refines its skill in generating realistic samples, and the discriminator enhances its ability to classify samples over time.



*Fig, 2.4 GAN Block Diagram*

The GAN framework can be mathematically formulated as a game of minimum and maximum values between the generator and discriminator networks. The generator's objective function is to decrease the probability of the discriminator accurately classifying the generated data as fake, while the discriminator's objective function is to increase the probability of correctly classifying both the real and generated data. This can be represented as:

|  |
| --- |
| **Min(G) Max(D) V (D, G)**  **V (D, G) =** |

The network labelled as G is responsible for generating data samples that are intended to resemble the real data distribution, while the network labelled as D is responsible for distinguishing between the real data and the generated data samples, x is a real data point drawn from the true data distribution pX, z is a random noise vector drawn from a prior distribution pZ, and G(z) is a generated data point obtained by applying the generator to the noise vector z. The first term in the objective function corresponds to the expected log probability that the discriminator correctly classifies a real data point, and the second term corresponds to the expected log probability that the discriminator incorrectly classifies a generated data point as real.

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.

Ⅴ. 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.

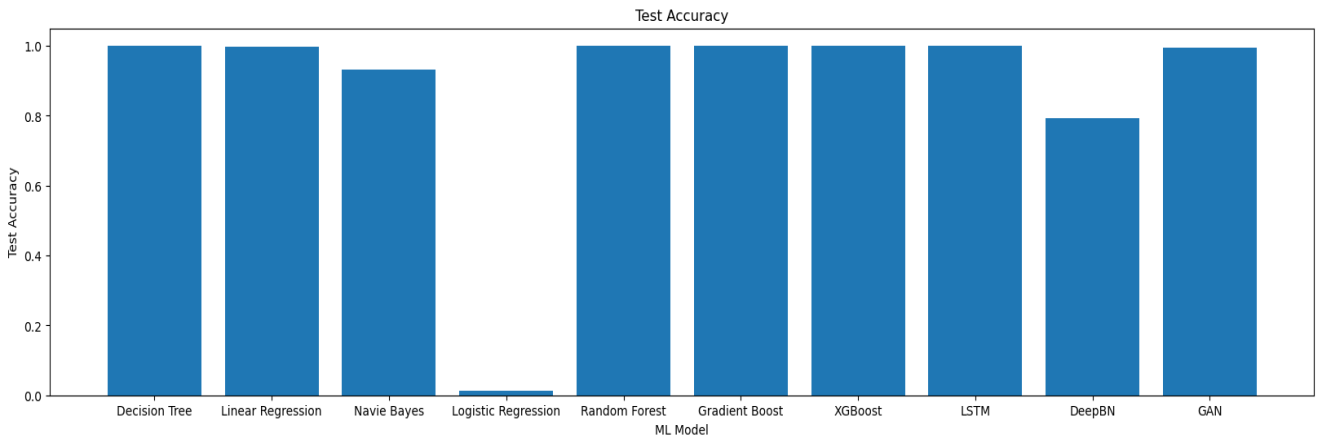
| **Classifier** | **Test Accuracy** | **Training Time** |
| --- | --- | --- |
| Decision Tree | 0.9962 | 10.846 |
| Linear Regression | 0.9775 | 5.988 |
| Navie Bayes | 0.9321 | 3.739 |
| Logistic Regression | 0.0141 | 376.921 |
| Random Forest | 0.9992 | 495.712 |
| Gradient Boost | 0.9594 | 6372.185 |
| 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*

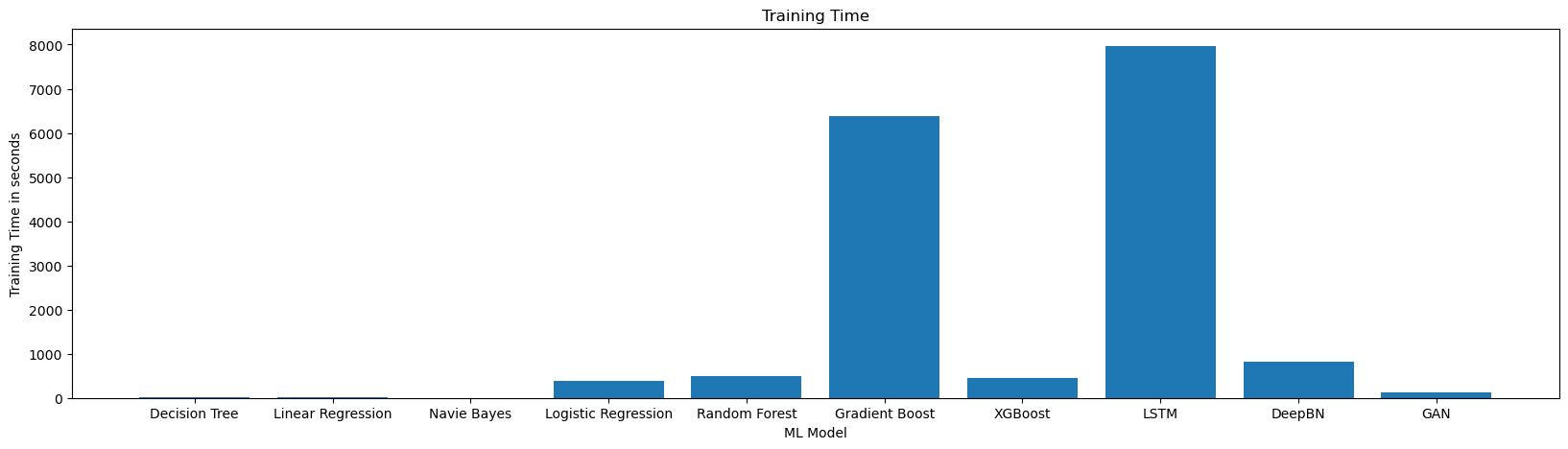
| **Epoch** | **Discriminator Loss** | **Generator Loss** | **Accuracy** |
| --- | --- | --- | --- |
| 1 | 38.05141 | 0.689847 | 42.19 |
| 100 | 0.131214 | 1.496475 | 100 |
| 200 | 0.096201 | 1.761345 | 100 |
| 300 | 0.025205 | 3.041839 | 100 |
| 400 | 0.011418 | 3.800014 | 100 |
| 500 | 0.005777 | 4.490452 | 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 scores more than 80. The bar graph says more about the training time required by all the classifiers.



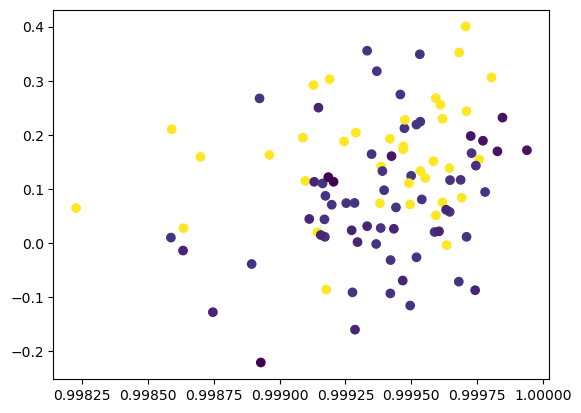
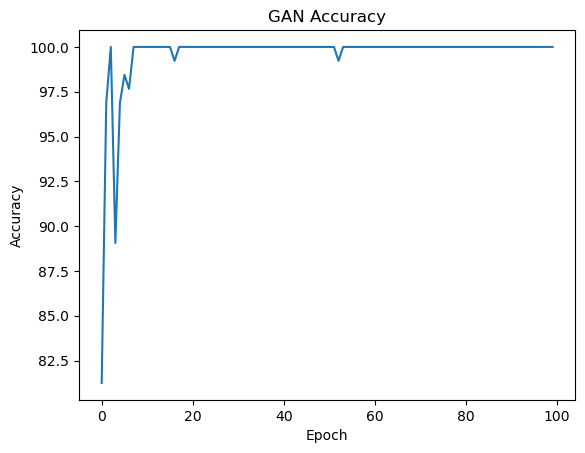
*Fig, 3.1 Testing Accuracy*



*Fig, 3.2 Training Time (in seconds)*

Ⅵ. 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 figures 4.1 and 4.2 shows a range of 1 to 0.99825. 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. 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.