Educational Administration: Theory and Practice
2024,30(4), 152-158
ISSN:2148-2403
https://kuey.net/

Research Article

A Cloud Based Honeycloud System For Malicious Detection Using Machine Learning Techniques

Kurra Chaitanya Kumar[1*], Busireddy Mokshitha Reddy[2], Naguru Tahaseen[3], Bhed Bahadur Bista[4], S.V.S. Ganga Devi[5]

[1*,2,3]BTech Student, Dept of C.S.E(Cyber Security), Madanapalle Institute of Technology & Science.
[4]Faculty of Software and Information Science, Iwate Prefectural University.
[5]Professor & Head, Department of C.S.E(Cyber Security), Madanapalle Institute of Technology & Science.

Citation: Kurra Chaitanya Kumar et al. (2024), A Cloud Based Honeycloud System For Malicious Detection Using Machine Learning Techniques. *Educational Administration: Theory And Practice*, *30*(4), 152-158,
Doi: 10.53555/kuey.v30i4.1425

ARTICLE INFO

ABSTRACT

With the massive expansion of IoT botnet DDoS attacks in recent years, IoT protection has now become one of the most concerned topics in the area of network security. In this paper, we propose a honeypot-based method that uses machine-learning techniques for malware detection in the IoT system. The IoT honeypot developed data is used as a dataset for the practical and dynamic training of a machine learning model. The honeypots are developed and placed in a cloud network that allows us to gather the unknown and known incidents in cloud computing. In our cloud network, we suggest a HoneyCloud system that highlights catching any attack or suspect action on protocols like Secure Shell (SSH) protocol, File Transfer Protocol (FTP), etc. These methods can be employed to define the distinction between Malicious and benign traffic. Also, we implemented various machine learning models in this paper and compared them with the parameters of True positive (TP) and false positive (FP) rates; this comparative analysis on which one of these machine learning-based classification algorithms would give us a low false positive rate.

Keywords: IoT, Distributed Denial of Service Attacks, Honeypots, Intrusion detection system, threats, Malwares, Cloud infrastructures.

1. INTRODUCTION

IoT, which is a network of connected things without human intervention, has also now become the source of propagating DDoS Attacks [1]. IoT machines can be more efficiently compromised than desktop computers. Thus, there has been significant growth in the occurrence of IoT-based botnet attacks. The botnet, directed to a network of bots, is the result of malware infections in an IoT network [2]. According to a recent survey, there are over 6 billion IoT devices on the planet; cybercriminals cannot easily overlook such a vast number of potentially vulnerable gadgets. A honeypot can be used as an intrusion detection mechanism that can mimic some or all of a server's behaviour, allowing server administrators to protect their servers by monitoring the capabilities of attackers. They are as they should be. The carrier may prevent denial-of-carrier (DoS) attacks to ensure reliable and uninterrupted service.

There are many ways by which a honeypot can be defined. A honeypot can be briefly described as an attacker's trap that mimics some or all of the activities of an actual device, and the attacker plays the real one. Honeypots can be flexibly deployed on the server side to most successfully respond to such attacks, protecting one's critical records and recording an attack through the ability to report malicious spores. Time can be traced. Honeypots can be generally divided into low-reactivity and high-reactivity honeypots [2]. Highly interactive honeypots simulate actual production structures along with bids and host distribution of responsibilities. They offer more protection and are more challenging to get hold of, but their economic price is relatively high. Alternatively, low-interaction honeypots emulate services that attackers may frequently request. They consume very few resources and can be easily maintained [3]. Both types of honeypots can be implemented as digital machines and hosted on the same physical server.

1.1 Honeypots

Since distributed denial of bearer attacks can undoubtedly be dangerous to the target server, it is essential to detect and mitigate such attacks. Although it is tough to protect yourself from attacks completely, several techniques have been suggested to deal with DDoS attacks. Two primary strategies for dealing with DDoS attacks include attack mitigation and identifying the source of the attack [4]. Honeypots can be used effectively in any situation. Effective detection and mitigation of bearer denial attacks is essential because they undoubtedly cause damage to the target server. Although it is tough to protect you from attacks completely, several strategies have been introduced to mitigate DDoS attacks. Two primary strategies for combating DDoS attacks are attack detection and mitigation. Honeypots are helpful in these techniques because of their versatility and versatility. A honeypot can mimic some or all of the capabilities of an herbal device, making it part of a truly isolated and carefully monitored community, albeit many miles away.

our proposed work, we used a honeypot framework to detect various malware system attempts on an IoT device. Data collected in the form of log documents can be used as input to get information about the version of the gadget we are using for educational purposes. The advantage of using honeypots on datasets to train versions is that we can learn the model through unknown versions of the malware family instead of using the simplest limited known data.

2. LITERATURE SURVEY

Machine Learning and its application in Honeypots

There are various honeypot-based methods present in the literature for protecting DDoS. The concept of the signature matching method has been used as a detection framework in some of these systems. Malware is detected on the basis of signatures obtained from their connected generated log files from the honeypot.

Many machine learning techniques have also been suggested to recognize DDoS based on the selection of statistical features using various supervised learning algorithms like SVM, Naïve Bayes, etc[5]

Abdullah et al. [2021] Research suggests implementing a commercial honeypot to attack botnet attackers by deliberately creating resources in the network to monitor and kill botnet attack behaviour carefully. The problem can be solved. As a result, content is reviewed and written to log files, quickly tagging this data with first-class accuracy and gaining insight into the actions involved in system usage. It is achieving increased productivity and improving the sustainability of smart factories. This review proposes a botnet detection model that combines honeypots and machine management specifically designed for smart factories. This vision is realized with hardware simulating the environment of an intelligent manufacturing unit.

Nasser Alsabilah et al. [2023] This paper proposes a unique technique to boost adaptive wavelet-based fully NIDS for intelligent local networks, mainly based on XGBoost combined with rough set theory. The proposed method is evaluated using the AUC-ROC curve, accuracy and F1 classification metrics, including the evaluation of unbiased registration factors. Experimental results highlight the effectiveness of our adaptive floating point-based NIDS by combining XGBoost with rough set theory expertise.

Alhan et al. [2022] It is stated that in a globalized world, people have become alienated from their social lives with digitization and have become addicted to age and that this addiction brings cyber threats and hence attacks; Alhan B suggested real-time detection of attacks. It is using HoneyPI and Machine Learning. What is achieved in this environment is a complete security architecture that integrates the HoneyPi into the Raspberry Pi, relying primarily on low-cost open software programs and hardware, making Naive Bayes a machine for the first time. It will be used as the learning algorithm that will be used for testing. This set of database rules is used because of its high degree of accuracy in very little information, and it is now done immediately by using it instead of painting through learning. So it is very cheap. Datasets IT procedures then use an LSTM set of machine learning rules that offer seamless and sequential operation.

Selvakumar Veluchamy et al. [2021] In this paper, Adaptive Deep Reinforcement Learning (ADRLH) ADRLH for Honeypots is completed to withstand internal and external DoS attacks. In a honeypot environment, the proposed DARLH tool implements DARL-based full-fledged IDS (Intrusion Detection System) vendors and Deep Recurrent Neural Network (DRNN)-based only IDS vendors to scan for various runtime DoS attacks. This strategy supports dynamic IDS against DOS attacks. Additionally, DARLH builds a closed poison distribution and server-side auditing system to keep monitoring events legitimate. Standard performance is assessed by fitting these images. The results were compared with existing structures consisting of GNBH, BCH, and RNSG.

Abyan Faishal Reza et al. [2022] This paper proposes the integration of a honeypot sensor (Suricata) in an SDN environment, i.e., the SD-Honeypot Community, to address DDoS attacks through device implementation techniques. The application uses various algorithms (Support Vector Machine (SVM), Multilayer Perceptron

(MLP), Gaussian Naive Bayes (GNB), K-Nearest Neighbours (KNN), Classification and Regression Trees (CART) and Random Forest (RF). And it is used comparatively. The dataset used in the analyzed simulation performed Internet Control Message Protocol (ICMP) flood records extracted from the Suricata sensor. They are ranking the effectiveness of the detection and mitigation modules. Several variables were tested to do this, particularly the accuracy, precision, negligence, and speed of the modern mitigation deployment approach. Forwards a flow rule change message for. Test results demonstrated the effectiveness of the CART ruleset in detecting and resolving interference.

Ruchi Vishwakarma et al. [2019] This paper presented a purely honeypot-based technique that exploits gadget reputation strategies to attack malware. The statistics generated by the IoT honeypot are used as datasets for realistic and dynamic training of the machine learning version. This approach is a more in-depth and practical start in the fight against zero-day DDoS attacks, which has now emerged as a sincere plan to protect the Internet of Things against DDoS attacks.

3. RESEARCH METHODOLOGY

While cloud computing is developing so rapidly, network management and security controls remain among the most significant issues for providers. In this case, automation, particularly the use of machine learning, is a rapidly growing approach to predicting and preventing safety risks and intimidations.

Machine learning (ML) is a subfield of "artificial intelligence" responsible for building the underlying computational system and developing a statistical model based entirely on proprietary data, known as "training data". There are four main types of learning in ML: supervised, unsupervised, semi-supervised, and Reinforcement learning.

The proposed device takes a proactive stance by creating a honeypot community, keeping in mind the early detection and evaluation of potential threats. This proactive approach complements the overall security posture of cloud-first systems.

3.1 Honeypot Deployment:

The project requires a strategic deployment of honeypots in cloud infrastructure. These honeypots mimic legitimate assets, enticing potential attackers by simulating correct assets and permitting their methods to be monitored and evaluated.

Diverse Honeypot Types: Various types of honeypots, including low-interaction and hyper-interaction honeypots, are used to simulate the extremes of interaction with potential attackers. This range provides a thorough understanding of the threat environment.

Cloud-Specific Honeypots: Tailored honeypots are designed to mimic cloud-based services and resources by impersonating cloud-based offerings and assets. This feature enables the identification of threats that may be specific to cloud computing.

Behavioral Analysis: The honeypot network uses advanced behavioral assessment techniques to infer malicious activity from legitimate interactions. This includes tracking access patterns, requests for statistics, and attempts to exploit capabilities.

Real-time Threat Intelligence: This project includes real-time threat intelligence to stay up-to-date on appearing threats. This ensures that the honeypot network evolves attack vectors and provides proactive protection against advanced cyber threats.

3.2 Automation using machine learning

Machine learning methods are used to automatically catch and jam malicious network visitors by analysing network traffic and recognizing patterns that indicate possible protection. Additionally, it mechanically detects and responds to suspect action on cloud assets by analysing asset usage patterns and identifying irregular patterns. When a distinctive pattern is seen, the proposed algorithm can mechanically take proper activity, including cancelling recognition of an event reaction. This can support groups save time and help by decreasing the need for manual monitoring and intervention.

3.3 SYSTEM ARCHITECTURE

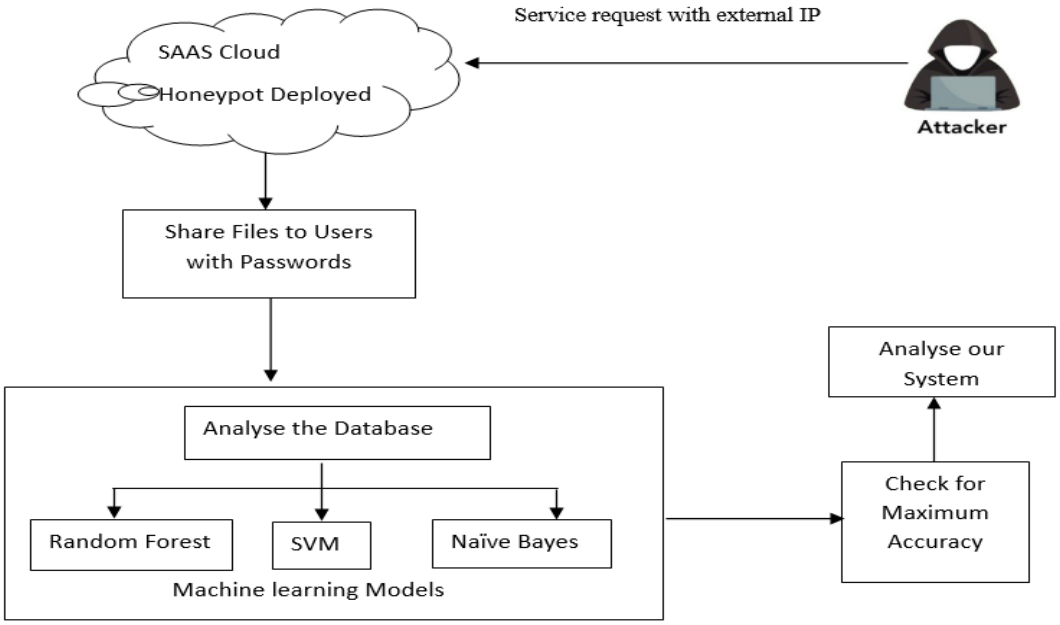

Fig.1 Proposed system architecture

Description about the system

Every machine that desires to communicate to a cloud instance originally transmits a standard the HTTP request. In the cloud network various VMs can be deployed in a honeynet and run on various operating systems. When requests are sent from one machine to another, packet streams are sent and obtained via the cloud's external IP. The data is gathered and delivered for future bat preprocessing. The data format is then provided as input to three machine-learning techniques of RF, SVM, and NBs to ensure higher accuracy.

BASIC Honeypot design

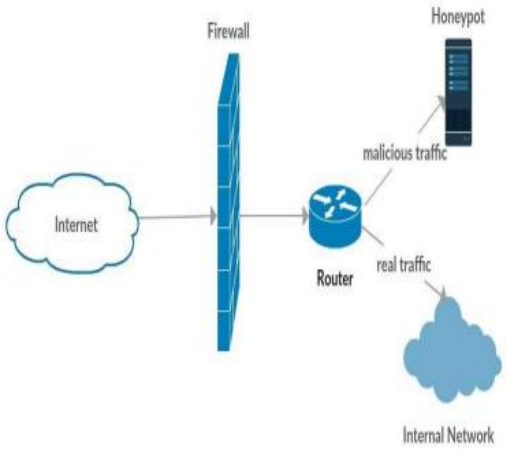

Fig.2 Basic Honeypot design

3.4 Machine Learning Module:

3.4.1 Naïve Bayes Classification Algorithm

Machine learning-based naive Bayes is a classification technique that believes that a selected feature in a data set is separated from other components. For example, take a fruit, especially an orange. Features such as colour (orange), shape (round), and diameter (three inches) are independent of other samples in the dataset. It increases the probability that the fruit is orange, hence the name Naive. Naïve Bayes plays better if the data is very long. Given that they're running a parallel map reduction on it, it's also easy to train. It creates it faster than the Support vector machine and Random Forest algorithms. Also, on our machine, when Naïve Bayes is introduced, it takes less time to learn and maintain the version.

3.4.2 Support Vector Machine (SVM):

In machine learning-based supervised learning, algorithm SVM is one of the algorithms that is more explanatory, compared to Naïve Bayes, which is a productive performance. It is based on a given function using $y = w.X + b$, where the weights (w) and bias (b) are calculated from the training data. Instead of underestimating the training error, SVM minimizes the generalization errors. Compared to overall performance, SVM outperforms Naïve Bayes because it constructs a hyperplane that increases the working margin. SVM generally takes longer to train than Naïve Bayes, but the forecasts are more precise.

3.4.3 Random Forest

Random Forest is an algorithm derived from the supervised algorithm and collected from decision trees (DT) algorithms. It is strong against overfitting and operates with numerical data. It emphasizes capabilities because it measures the impact of each predictor on the outcome. The major drawback of this algorithm is that the increased variety of trees can cause the computation and training procedure to be slow and ineffective for real-time forecasts. The random forest algorithm provides better accuracy compared with the Naïve Bayes and SVM.

4. PERFORMANCE EVALUATION METRICS

Evaluation metrics provide positive feedback to the model. The model improves with each round based primarily on the evaluation metrics for the validation and test sets obtained in the previous game. The goal of using these metrics is to create a version that adequately responds to out-of-sample data styles.

Accuracy $= (True\ positives + True\ Negatives)/ (True\ positives + True\ negatives + False\ positives + False\ negatives)$

Precision (P): Precision is the fraction of the correct tags generated by the NER to the total number of tags generated.

$$Precision\ (P) = True\ Positive/True\ Positive + False\ Positive$$

Recall(R): Recall is the fraction of the correct tags generated by the NER to the total number of correct tags.

$$Recall\ (R) = True\ Positive/True\ Positive + False\ Nagative$$

• **F1-Score:** F-score is the weighted harmonic mean of precision and recall.

$$F1 = 2 * (precision * recall) / (precision + recall)$$

4.1 Honeypot Simulation

In the cloud network, once the honeypots deployed instance, we begin employing the external IP of the cloud network with the correct port variety within the browser. Where our honeypots are deployed, there we make two examples here: one is the attacker instance and the the victim instance. And create an SSH key to share the facts. In our cloud example, we simulate both the attacker's and the victim's devices are present the results. An attacker sample may be floating around the Metasploit Framework, where we can try different exploits on our complex cloud instance.

4.1.2 IoT Virtual Honeypot:

The first step in our proposed technique is to knowingly enable attackers to exploit a vulnerability in IoT devices. To emulate this behavior, we want a machine or tool that can act precisely like a usable IoT device, enabling malicious rotation without allowing an attacker to consider the reality of the exploit. As discussed above, depending on the extent of interaction, honeypots are labeled as high interaction honeypots (HIH), low interaction honeypots (LIH), and a mixture of each, medium interaction honeypots (MIH). Since installing a highly interactive honeypot (HIH) for functional, resource-constrained IoT devices is impossible, choosing a medium-interactive honeypot (MIH) may be better than the other two. That's why it's called an IoT 'virtual' honeypot: in this case, we'll implement it by emulating an IoT platform using the IoT verbal exchange protocol—network site visitors, payload, malware samples, the toolkit used by the attacker, etc. Attack techniques, e.g., it can be saved with a honey pot. Here is a list of some recently developed IoT honeypots for DDoS detection.

A more viable IoT honeypot should now be able to simulate IoT devices by agreeing on multiple communication protocols and infecting the entire IoT platform with all supported application layer protocols. It needs to be successful. Some of the famous supporting protocols used for IoT voice transformation are MQTT (Message Queue Telemetry Transport), XMPP (Extensible Messaging and Presence Protocol), and AMQP (Enhanced Message Queue Protocol), developed by IBM for instant messaging (IM). Provide convenience. An economic industry originator, CoAP (Constrained Application Protocol) - is a set of protocols used to invent UPnP (Universal Plug and Play) network devices and HTTP REST, which is helpful with limited resources and high speed. They are designed for tools. REST is an architectural style commonly used in machine-to-machine (M2M)

voice exchange and IoT architectures. We will use ThingPot to prevent many malware attacks from all the honeypots above.

4.2 RESULTS AND DISCUSSION

The collected data must first go through a training phase. In this segment, to get the proper effect from honeypots, we allow the machine learning to rendering to daily HTTP protocol requests that are not attacks. The training phase is located in a secure environment. Games will be labelled affording to the "message" discipline in the dataset. After training the system in WEKA conferring to the chosen features, the machine learning methods are executed. This is an excellent way to achieve maximum accuracy.

Table.1 Accuracy Comparison of Random Forest Performances with deference to honeypots in HoneyCloud

Machine learning Models	True Positive	False Positive	Precision	Recall	F1-Score
SVM	0.98	0.108	98.75	85.55	91.67
Naïve Bayes	0.97	0.109	97.80	90.25	93.87
Random Forest	0.98	0.1	99	98.50	98.74

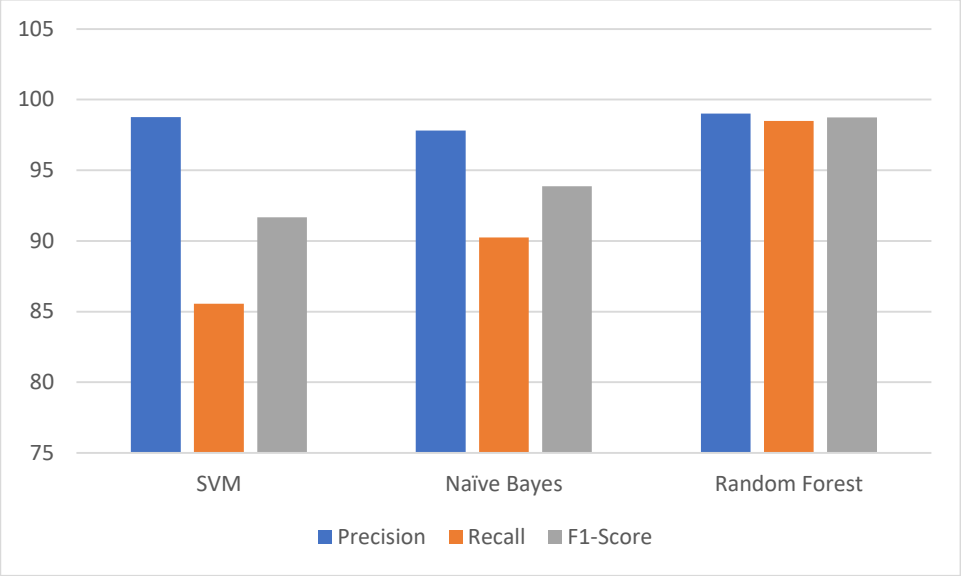

Fig.3 Performance evaluation between various machine learning methods

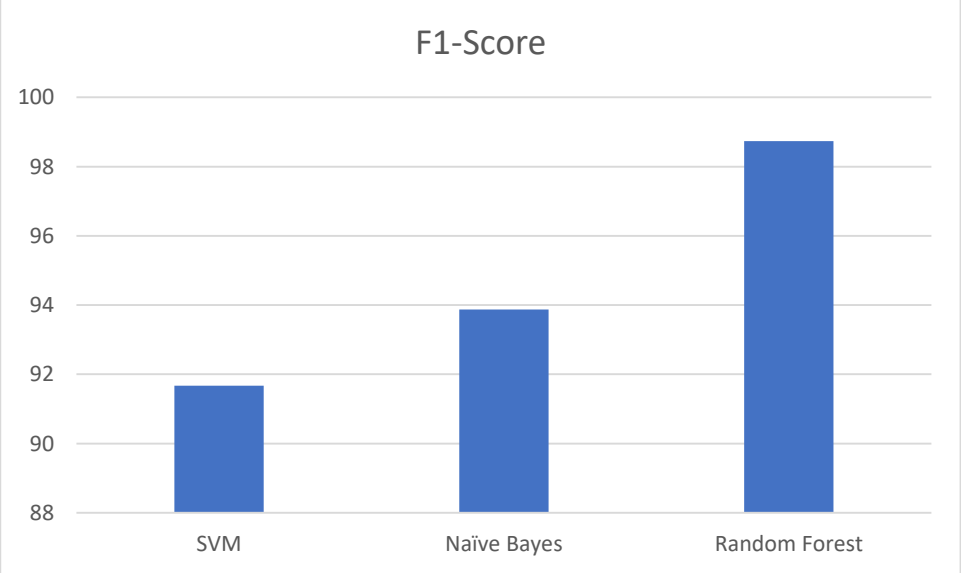

Fig.4 Comparison of machine learning performance in F1-Score

5 CONCLUSION

In this paper, we presented a HoneyCloud system consisting of a robust honeypot for identifying attacks. To review the logged details actions by this Honeypot, we tested with three various machine-learning techniques. We have implemented the machine learning models are Naïve Bayes (NB), Support vector machine (SVM), and Random Forest (RF) so that new incoming data from these honey clouds can be classified as malicious with more incredible accuracy. The experimental results in the paper show that the Random Forest technique provides higher accuracy.

In future work, we will concentrate on reducing false negative rates by better association with data from the test conditions. We also intend to add other features to automate the procedure more effectively to reduce human efforts and increase precision.

REFERENCES

1. C. Huang and J. Liu, “Automatic identification of honeypot server using machine learning techniques,” Security and Communication Networks, vol. 2019.
2. A. Abdullah and S. H. Kok, 2021, “Honeypot Coupled Machine Learning Model for Botnet Detection and Classification in IoT Smart Factory – An Investigation,”, p. 04003.
3. A. Alhan and E. N. Yilmaz, 2022, “Real-Time Cyber Attack Detection Over HoneyPi Using Machine Learning,”, pp. 1394–1401.
4. V. Mehta and S. Rajpoot, 2015, “Threat prediction using honeypot and machine learning,”, pp. 278–282, 2015.
5. Selvakumar Veluchamy and Ruba Soundar Kathavarayan, 2021, “Deep Reinforcement Learning for Building Honeypots against Runtime DOS Attack”.
6. Azween Abdullah and S.H. Kok, 2021, “Honeypot Coupled Machine Learning Model for Botnet Detection and Classification in IoT Smart Factory”, pp.1-14.
7. Fauzi Dwi Setiawan Sumadi, Alrizal Rakhmat Widagdo, 2022, “SD-Honeypot Integration for Mitigating DDoS Attack Using Machine Learning Approaches”, pp.39-44.
8. Ruchi Vishwakarma and Ankit Kumar Jain, 2019, “A Honeypot with Machine Learning based Detection Framework for Defending IoT based Botnet DDoS Attacks”, pp.1019-1024.
9. Hatice Beyza and Mehmet Ali, 2021, “Password Attack Analysis Over Honeypot Using Machine Learning Password Attack Analysis”, pp.388-402.
10. Nasser Alsabilah, Danda B. Rawat, "An Adaptive Flow-based NIDS for Smart Home Networks Against Malware Behavior Using XGBoost combined with Rough Set Theory", pp.15-22, 2023.
11. Atharva Auti; Shrawani Pagar; Vivek Mishra, “Honey Track: An improved honeypot”,
12. Brown, M. "Machine Learning for Anomaly Detection in Network Security: A Comprehensive Review."
13. Davis, S. "Dynamic Threat Intelligence Feeds: Enhancing DDoS Attack Detection in IoT Environments."
14. White, D. "Automated Mitigation Strategies for IoT-based Botnet DDoS Attacks: A Machine Learning Approach."