Strengthening Security: Machine Learning-Powered Honeypot Dashboard for Brute Force Attacks Detection

Nur Fatin Nabilah Md.Irzan 1, a) and Siti Husna Abdul Rahman1, 2, b)

1Faculty of Computing and Informatics, Multimedia University, Jalan Multimedia, 63100 Cyberjaya, Malaysia.

2Centre for Cybersecurity and Quantum Computing, CoE for Artificial Intelligence, Multimedia University, Jalan Multimedia, 63100 Cyberjaya, Malaysia.

*a) Corresponding author: nnabilahzan@gmail.com  
b)siti.husna@mmu.edu.my*

**Abstract.** The increasing number of cyber-attacks has been a concern where particularly in Malaysia, the number of intrusion attempts has increased over years. Intrusion attempts or brute force attacks work by exploiting weak authentication mechanisms that may lead to data breaches, system compromise, and operational disruptions. Traditional security tools are no longer sufficient to control these activities as these attacks continuously evolve and adapt to new security mechanisms. To address this issue, this research project explores the integration of honeypot and machine learning to enhance the detection and analysis of brute force attacks. A medium interaction honeypot, Cowrie, is deployed as a decoy system while machine learning model, Random Forest (RF) and Decision Tree (DT) is chosen to train and classify the honeypot logs data that contained brute force attack. Additionally, to provide actionable intelligence, a honeypot dashboard will be developed to display classification results, attacks patterns, and trends, enabling security teams to gain deeper insights into brute force attack behaviors. This research aims to demonstrate the effectiveness of integrating honeypots and machine learning in improving intrusion detection systems, contributing to stronger authentication mechanisms and overall network security.

# introduction

A number of cyber cases have been increasing frequently day by day which exposes risks and various threats to both individuals and organizations. These threats often led to serious consequences, such as unauthorized access to sensitive information, identity theft and data breaches. A most common example of attacks, Brute Force attack, has been a persistent form which exploits the weak authentication protocols of users’ passwords or confidential. In a conceptual view, this attack provides a simple workflow but can result in a bigger impact such as data leakage, operational disruption, system compromise [1]. Figure 1 shows how attackers use brute force techniques to exploit the victims’ credentials.

One of the key challenges in the field of cybersecurity is to accurately detect and analyze attack data to generate actionable intelligence. However, getting insights into the threat identities, motivations and resources remains a complex task [2]. To address this issue, honeypots have proven to be an effective tool for detecting cyber threats by simulating them as a real server to attract malicious attackers. This tool captures all attacker activities while keeping the actual server environment secure [3].

A screen shot of a computer

Description automatically generated

**FIGURE 1.** Brute force attack flow

Nonetheless, honeypot data is typically stored in a log format which is not user-friendly. Managing and analyzing this log data would be time consuming and inefficient for the network administrator, especially when handling big size data and identifying attack patterns and behavior manually. Additionally, current methodologies for evaluating the honeypot performance are insufficient thus limiting the ability to derive a meaningful insight from collected data. Leveraging this data for predicting future attacks and enhancing organizational security also remains a challenge [4]. Traditional detection methods, such as static input validation and pattern matching, frequently fall short in adapting to the dynamic nature of modern cyber threats [5].

This project addresses this gap by exploring the integration of machine learning with honeypots to enhance the detection of brute force intrusions. By utilizing machine learning models to classify and analyze honeypot data, this project aims to improve upon traditional methods and offer more effective, data-driven insights for understanding and mitigating brute force attacks.

The objectives of this project are as below:

* To deploy a honeypot system as a vulnerable system for the attackers to launch brute force attack
* To evaluate the effectiveness of the deployed honeypot in detecting brute force attack with machine learning
* To analyze the data captured by the honeypot focusing on brute force attack patterns and characteristics

The significances of this research are as follows:

* This research demonstrates the effectiveness of honeypot system for attack detection through simulation
* This research enhances understanding of how machine learning and honeypot can contribute to cybersecurity field especially for brute force attack detection
* This research improved the traditional honeypot system with combination of machine learning algorithms
* The results and findings from this research would provide valuable insights into brute force attacks patterns and characteristics which can be used to strengthen security policies
* This research serves as a reference for future studies in intrusion detection and threat intelligence.

# RELATED WORK

Brute force attacks continue to pose a significant threat to information systems, often targeting services such as SSH to gain unauthorized access. To counteract these risks, organizations are increasingly integrating advanced monitoring tools as part of their cybersecurity strategies. Tools such as Intrusion Detection Systems (IDS) and honeypots play a critical role in the early detection of malicious activities, including repeated failed login attempts that may indicate a brute force attempt. Early identification of such anomalies can help prevent high-risk security breaches. Unlike traditional detection systems, honeypots deliberately attract attackers, allowing organizations to observe and record adversarial behavior in a controlled environment. This not only reveals the attackers’ techniques and motives but also enables security teams to develop more informed and proactive defense mechanisms. Furthermore, the deployment of these monitoring tools facilitates continuous observation of system logs and network traffic, enhancing an organization’s ability to detect, analyze, and respond to evolving threats in real time.

Paper [6] deployed a honeypot in cloud environment for 30 days. The honeypot chosen was Dionaea honeypot which focuses on malware attacks. The results collected then is compared with the global results from multiple honeypots which aims to find the differences method used and type of attacks in Asia region especially Japan. Another paper from [7] designed a new innovation by integrating multiple security mechanisms that are Snort IDS, Dionaea, firewall, audit watch, honey folder, SDN controller and Complex Event Processing to detect ransomware attacks.

Study from [8] uses a high interaction honeypot T-Pot to simulate the effectiveness of honeypot against cyberattacks. The deployed honeypot was observed for more than six months in the faculty of computer science and electronics in an institutional campus. Results from the study showed that Denial of Services is highly detected with 39%, and followed by the SQL Injection attack 21%, Cross-site scripting (XSS) 12% and 10% of Brute Force Attacks.

A study by [9] implements few methods of machine learning classifiers to detect brute force attacks on a dataset of CSE-CIC-IDS 2018 that focuses on SHH and FTP services. Some of the methods algorithms used are Naïve Bayes, Decision Tree, Random Forest, and Logistic Regression. As a result, Gaussian Naïve Bayes (GNB) was considered effectives for handling the datasets but has limitation regarding the feature independence while Logistic Regression (LR) provided a faster classification but do struggle with nonlinear relationship. Thus, it is concluded that Random Forest (RF) algorithm provides the highest accuracy for the SSH and FTP brute force detection results.

# METHODOLOGY

The overview of this research employs the following processes as shown in Figure 2. below. We identify three main components that are important in this research: honeypot deployment environment, machine learning models training and testing for classification, and also honeypot dashboard. These components primarily contribute to the detection and visualization of brute force attacks patterns in this study.

A diagram of a system

Description automatically generated

**FIGURE 2.** Research phases

## Honeypot Deployment Environment

The honeypot used for this research is a medium interaction open-source honeypot known as Cowrie honeypot. This honeypot is deployed in a virtual machine using Oracle VM VirtualBox in an Ubuntu based together with Kali Linux as an attacker side as presented in Figure 3. A sample of the honeypot captured logs file can be seen in Figure 4. This honeypot aims to capture network traffic, especially ssh attempts. A low-to-medium interaction honeypot is more suitable to be deployed for this project as it is used in a small testing environment while high interaction honeypot is more complex and expensive to implement [8]. In addition, the likelihood of infection from malware or hacking by the attacker is greater with high-interaction honeypots compared to low-interaction honeypots [6]. This low-to-medium honeypot is still able to capture attacks and provide the sufficient data that is needed for this project making it ideal for identifying and documenting brute force attacks without exposing the system to high levels of risk. Logs included are such as Ip address and ports source, timestamp, targeted port, login credentials, session interactions and activities executed.

|  |  |
| --- | --- |
| A diagram of a computer network  Description automatically generated  **FIGURE 3.** Simulation setup | A screenshot of a computer screen  AI-generated content may be incorrect.  **FIGURE 4.** Honeypot logs |

## Machine Learning Models Training and Testing

We implemented a supervised method which are Random Forest (RF) and Decision Tree (DT) to classify and analyze the attacks data. By integrating machine learning in this project, we are able to process the datasets, detect patterns and classify data effectively. In addition, Random Forest has been tested to be the best accurate algorithm for classifying data and detecting SSH brute force attacks.

In this project, the dataset of Cowrie login activities was used to train on both machine learning models, Random Forest (RF) and Decision Tree (DT) to detect honeypot brute force attacks and to evaluate their effectiveness in brute force attack detection in classification and overall performance. The activities were classified into two categories of normal traffic and brute force attack. The number of logins attempts were analyzed with focus on failed logins that occurred multiple times within a short time period. IPs with a high number of failed attempts within a short timestamp is labelled as brute force attack. Login bursts occurring within defined time windows were also considered malicious. The objective of this labelling is to detect patterns associated with brute force behavior and enhance the detection capabilities. This dataset was split into 80% for training and 20% for testing to evaluate the models' accuracy, precision, recall, and F1-score effectively.

• TP (True Positive) = Attacks correctly classified as brute force

• TN (True Negative) = Traffic with no attacks correctly classified as normal

• FP (False Positive) = Traffic with no attack incorrectly classified as brute force

• FN (False Negative) = Attacks incorrectly classified as normal

Accuracy in Equation (1):

(1)

Precision in Equation (2):

(2)

Recall in Equation (3):

(3)

F1 Score in Equation (4):

(4)

## Honeypot Dashboard Development

In support of these findings, a web-based dashboard was developed to visualize the honeypot data and brute force detection results. Built using Flask for the backend and Chart.js with standard web technologies for the frontend, the dashboard offers a clear and interactive interface to monitor key metrics such as total login attempts, failed versus successful logins, top attacking IP addresses, and attack trends over time. It also displays the latest suspicious events in a tabular format, providing a quick overview of recent activities. The dashboard transforms raw log data into informative visuals that are easy to understand and navigate, which helps users and analysts stay informed about potential threats without digging through complex log files.

## RESULTS AND DISCUSSION

After training and testing the datasets with two different models, we found that both classifiers Decision Tree and Random Forest models performed well in detecting brute force attacks on the honeypot datasets. The accuracy for Decision Tree has achieved 0.99 and scored a perfect 1.00 in precision, recall, and F1-score. While Random Forest model similarly achieved accuracy of 0.99 with only lightly difference in precision, recall, and F1-score values, 0.99, as shown in Table 1. The results obtained from these evaluations have shown that these models are highly capable of classifying the pattern accurately.

**TABLE 1.** Models performance evaluation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine Learning Models** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
| Decision Tree | 0.99 | 1.00 | 1.00 | 1.00 |
| Random Forest | 0.99 | 0.99 | 0.99 | 0.99 |

Although these results may differ from common findings, Random Forest has been claimed in existing literature to be the best for classifying brute force attacks, but decision tree is found to be lead is due to several factors that related to the nature of the dataset itself. This dataset consists of a record of ssh login attempts where it is clearly defined patterned which allow decision tree to overfit and create effective splits and classify the data with high accuracy without the ensemble need of random forest algorithm. This overfitting could also cause the model to memorize the training data and achieve near-perfect scores during testing. Moreover, the labelling method used for this dataset may also have been aligned with the rule based splitting mechanism of Decision Tree which contributed to a higher performance. The results of honeypot dashboard integrated with machine learning is as illustrated in Figure 5 and Figure 6.

Further insights into the confusion matrix as shown in Figure 7 and Figure 8, for Decision Tree classifier a number of 103,303 detected as true positives, 1033 of true negatives with 243 false positives and 231 false negatives, while Random Forest classifier detected 103,505 true positives, 401 of true negatives, 875 false positives and 56 false negatives. From here we can see that Decision Tree was more inclined to classify events as a brute force attack where both true positives and false positives are considered high. Random Forest on the other hand reduced the number of false negatives demonstrating better precision. Relating to the real-world situation particularly in cybersecurity, the fewer false negatives generated such as Random Forest are more desirable as missing an actual attack could expose the system to be exploited and cause damage. However, having a high number of false positives could lead to wasted resources and alerts which is inefficient.

|  |  |
| --- | --- |
| **FIGURE 5.** Honeypot dashboard **FIGURE 6.** Honeypot dashboard | |
|  | |
|  | |
| A graph of a forest confusion matrix  AI-generated content may be incorrect.  **FIGURE 7.** Random forest confusion matrix | A graph of a tree confusion matrix  AI-generated content may be incorrect.  **FIGURE 8.** Decision tree confusion matrix |

As presented in Figure 9 the important features that contribute to detecting brute force attacks which include:

* Session connects count: The number of source IP established for ‘cowrie.session.connect’
* Failed login attempts: The number of failed attempts tht represents by ‘cowrie.login.failed.’
* Successful login attempts: The number of ‘cowrie.login.success’ of IPs
* Total login attempts: The total number of failed and successful login attempts
* Timestamps: To identify and observe the login patterns and behavior

A graph with blue rectangles

AI-generated content may be incorrect.

**FIGURE 9.** Important features

# CONCLUSION and future work

This research has presented the integration of honeypot and machine learning for detecting brute force SSH attacks, where attackers attempting to gain unauthorized access to victim’s account or device. We also have developed a dashboard to enhance the monitoring and management system of such attacks. This study involved analyzing the logs from Cowrie honeypot and training the data using two supervised machine learnings, Decision Tree and Random Forest, to classify the network traffics from honeypot logs into normal and brute force attack categories.

Results show that the honeypot is effective in capturing brute force attack data. While the Decision Tree model is shown to outperform with a slightly higher accuracy than Random Forest in the brute force attack classification with cowrie honeypot logs data. While the components have been extensively studied together, the strength of this research lies in its end-to-end implementation. This approach presents a lightweight, cost-effective and a reproducible framework that is suitable for a small to medium scope like academic and training environment.

Additionally, this research is limited to logs from a single honeypot type and is focused on SSH Brute force scenarios. For future research, we recommend several enhancements to be made by expanding the dataset and exploring various attacks vectors to improve the system’s scalability, implementing distributed number of honeypots with different targets would provide more varied data. Adding prevention steps by linking the system with security event management tools so that it enables automated alerts for more effective threat response.

# References

1. Kumari, A., D. Gupta, and M. Uppal, in *2024 Asia Pacific Conference on Innovation in Technology (APCIT)* (IEEE, Mysore, India, 2024), pp. 1–5.
2. Zobal, L., D. Kolář, and R. Fujdiak, “Current State of Honeypots and Deception Strategies in Cybersecurity,” in *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)* (IEEE, Dublin, Ireland, 2019), pp. 1–9.
3. Shivaprasad, T., A. S. Moulya, and N. Guruprasad, “Enhancing Network Security through a Multi-layered Honeypot Architecture with Integrated Network Monitoring Tools,” in *Proceedings of the 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)* (IEEE, New Delhi, India, 2024), pp. 473–477.
4. Sokol, P., and A. Gajdoš, “Prediction of Attacks Against Honeynet Based on Time Series Modeling,” *Advances in Intelligent Systems and Computing* **662**, 360–371 (2018).
5. M. B. A., K. A. Shastry, M. M., and Aravind, “Intelligent Defense Strategies: Machine Learning-Enhanced SQL Injection Detection and Prevention via Honeypots,” in *Proceedings of the International Conference on Intelligent Algorithms and Computational Intelligence Systems (IACIS)* (Hassan, India, 2024), pp. 1–7.
6. Saikawa, K., and V. Klyuev, “Detection and Classification of Malicious Access using a Dionaea Honeypot,” in *Proceedings of the IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)* (IEEE, Metz, France, 2019).
7. Chakkaravarthy, S., D. Sangeetha, M. V. Cruz, V. Vaidehi, and B. Raman, “Design of Intrusion Detection Honeypot Using Social Leopard Algorithm to Detect IoT Ransomware Attacks,” *IEEE Access* **8**, 169944–169956 (2020).
8. Martínez, S., H. Moreno, and M. Hernández, “Analysis of Intrusions into Computer Systems using Honeypots,” *International Journal of Intelligent Systems and Applications in Engineering* **11**(6s), 461–472 (2023).
9. Hamza, A., and R. Al-Janabi, “Detecting Brute Force Attacks Using Machine Learning,” *BIO Web of Conferences* **97**, 00045 (2024).