AI-Driven Intrusion Detection System Using SSH Honeypots

Prof. Swapnil Chaudhari[1], Abhishek Satpute[2], Vishwajit Gaikwad[3] and Yash Kakade[4]
[1] *School of Computing, MIT ADT University, Pune 412201, Maharashtra, India*

***Abstract*—With the rapid evolution of cyber threats targeting critical services like SSH, traditional Intrusion Detection Systems (IDS) are often unable to handle zero-day attacks and advanced persistent threats. This work proposes an intelligent IDS powered by SSH honeypots combined with machine learning. The honeypots simulate vulnerable SSH services to capture attacker behavior, which is then analyzed using Random Forest classifiers and Autoencoders for accurate intrusion detection. Our AI-based framework shows robust detection rates across multiple attack vectors, offering dynamic adaptability to evolving threats. The proposed system demonstrates a promising defense mechanism, bridging the gap between traditional signature-based systems and modern AI-driven security solutions.**

Index Terms*— *Intrusion Detection System (IDS), SSH Honeypot, Machine Learning, Anomaly Detection, Cybersecurity

I. INTRODUCTION

Secure Shell (SSH) is one of the most important protocols for remote server management in the modern digital world. But because of its significance, it is also frequently the target of cyberattacks, such as malware injections, brute-force login attempts, and credential theft. Conventional intrusion detection systems (IDS) frequently use signature-based detection, which has trouble spotting emerging threats. There is an urgent need for more intelligent, flexible security measures as cyberattacks get more complex.

By serving as decoy systems intended to draw in and keep an eye on attackers without endangering actual assets, honeypots—especially SSH honeypots—offer a practical solution. They offer insightful information about the methods, tools, and behaviour of attackers. However, it takes a lot of time and is inefficient to manually analyse large amounts of honeypot data.Honeypots, especially SSH honeypots, offer an effective solution by acting as decoy systems designed to attract and monitor attackers without risking real assets. They provide valuable insights into attacker behavior, tools, and techniques. However, manually analyzing large volumes of honeypot data is time-consuming and inefficient.

This project proposes an AI-driven Intrusion Detection System that integrates SSH honeypots with Machine Learning techniques. By capturing attacker interactions and analyzing them with supervised (Random Forest, Decision Trees) and unsupervised (Autoencoder) learning models, the system can detect known and unknown threats in real time. Our approach aims to build a dynamic, scalable, and highly accurate defense system against modern cyberattacks.

Intrusion Detection Systems (IDS) are critical in identifying unauthorized activities within computer networks. With the increasing sophistication of cyberattacks, traditional signature-based systems struggle to detect new, evolving threats. To enhance early detection capabilities, this work integrates SSH honeypots with Artificial Intelligence (AI) methods. SSH honeypots act as deceptive traps, attracting attackers and recording their activities, which can then be analyzed using Machine Learning (ML) models for real-time threat identification.

1.1 SSH Honeypots for Attack Capture

Usage of honeypots has proved to be a viable method of intelligence collection about the behavior of attackers. An SSH honeypot is a server designed to appear like an honest SSH service but is made vulnerable on purpose, which interacts with attackers without jeopardizing real systems.
As explained by Spitzner (2003), an SSH honeypot setup usually includes:

1. Simulated Authentication: Logs in using dummy or weak credentials.
2. Command Interaction: Captures all commands run by the attacker after authentication.

In most configurations, a high-interaction honeypot is used (Fig. 1.1), which allows for complete session monitoring, including file transfers and network activity. These interactions create a rich data set for later analysis.

Figure 1.1 Honeypot model recording attack sessions. New honeypots such as Cowrie and Kippo provide improved functionality by mimicking actual file systems, reacting to attacker input, and even simulating outbound connections. This makes attackers think they have taken over an actual machine, prompting further infiltration attempts, which are logged for analysis.

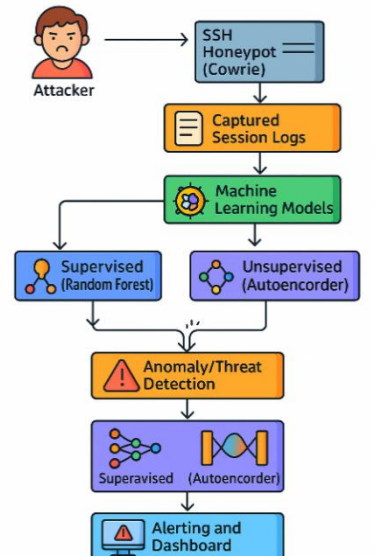

Figure 1 SSH diagram

1.2 Machine Learning-Based Attack Detection

- Once honeypot data is collected, Machine Learning models are employed to analyze and detect anomalies. Unlike traditional IDS that rely on fixed signatures, ML models learn patterns from data and adapt to detect unseen attacks.
- The system uses two main AI strategies:
- Supervised Learning Models: These models, like Random Forests and Decision Trees, are trained on labeled datasets containing examples of both normal and malicious sessions. They predict the probability of an incoming session being an attack.
- Unsupervised Learning Models: Autoencoders and anomaly detection techniques are used where labeled data is scarce. These models learn the normal behavior and flag deviations as potential intrusions.
- Formally, supervised learning aims to find a function f:X→Yf: X \rightarrow Yf:X→Y that maps input features XXX (session data) to output labels YYY (attack or normal).
  In unsupervised learning, the objective is to model the distribution of the data p(X)p(X)p(X) and detect instances that significantly deviate from learned normal behavior.
- Over multiple training iterations, the models refine their boundaries to maximize correct predictions while minimizing false positives and negatives. This AI-driven framework thus supports the detection of new types of attacks that traditional IDS might miss.
- The versatility of AI models has given rise to several architectural innovations and improvements:
- Random Forests: Ensemble of decision trees to improve robustness.
- Gradient Boosted Trees: Focus on hard-to-classify sessions.
- Autoencoders: Compress and reconstruct data to detect anomalies based on reconstruction errors.
- Isolation Forests: Detect outliers effectively in large datasets.
- These techniques are crucial for processing the large, noisy, and diverse datasets collected by honeypots.
- PGGAN: Used progressive growing for high-resolution image synthesis.

1.3 Real-Time Attack Detection and Alerting Process

An essential component of the proposed system is the real-time detection and alerting process, which ensures immediate response to active threats.

The real-time process involves:

- Log Collection: SSH honeypots constantly generate event logs from attacker activities.
- Feature Stream Processing: New session data is immediately preprocessed into feature vectors.
- Model Inference: The trained ML models predict in real-time whether a session is malicious or benign.
- Threat Scoring: Based on the model's output probabilities, a dynamic threat score is assigned.
- Alert Generation: If the threat score exceeds a pre-defined threshold, an automatic alert is

triggered to the Security Operations Center (SOC).

The dynamic sco ring formula is:

Threat Score=α×Login Attempts+β×Command Entr opy+γ×Anomalous Behavior Score

This real-time detection pipeline ensures minimal delay between attack identification and response, critical for mitigating fast-moving threats like credential harvesting or malware deployment.

II. RELATED WORK

2.1 Traditional Intrusion Detection Systems

Signature-based IDS tools like Snort and Bro (Zeek) are the stalwarts of cybersecurity protection. They keep extensive libraries of known attack signatures and raise an alarm when patterns match. But they are plagued by high false-negative rates when confronted by unknown or obfuscated attacks.

2.2 Honeypot Technologies

Honeypots, originally promoted by projects such as Honeyd and more recently extended to platforms such as Cowrie and Dionaea, are created to mimic vulnerable services to mislead attackers. SSH honeypots, in particular, impersonate SSH services to entice brute force and exploitation attacks. Honeypots are superior for information gathering, yet most deployments have no built-in intelligence for proactive threat blocking.

2.3 Machine Learning for Intrusion Detection

Recent advances have seen the application of machine learning techniques to cybersecurity, using algorithms such as Support Vector Machines (SVM), Decision Trees, Random Forests, and Deep Neural Networks to detect anomalies in network traffic. However, many studies rely on outdated datasets (e.g., KDDCup'99) that do not accurately represent modern attack landscapes, limiting the generalizability of their results.

2.4 Gap Analysis

Few studies focus specifically on SSH-based attacks using real-world data. Furthermore, existing systems often treat detection and data collection separately, leading to delayed responses. Our work aims to integrate these components into a unified, real-time AI-driven defense system.

III. METHODOLOGY

3.1 SSH Honeypot Deployment

We deploy Cowrie, a low-interaction SSH honeypot capable of simulating a full SSH service. The honeypot records detailed logs of connection attempts, authentication trials, commands executed, and file transfer attempts. Multiple instances are placed on public IP addresses to attract diverse attack vectors

3.2 Data Capture and Storage

Captured logs are automatically parsed and stored in a centralized NoSQL database (MongoDB). Each session entry includes:

- IP address
- Timestamp
- Login credentials attempted
- Commands issued
- Session duration
- Download/upload attempts

3.3 Feature Extraction

We design a feature extraction pipeline that processes the raw session data into structured feature vectors suitable for machine learning. Key features include:

- Number of login attempts per session
- Login success/failure rates
- Entropy of entered commands
- Command sequence length
- Average time between commands
- Usage of known malicious binaries (wget, curl)

3.4 AI-based Intrusion Detection

The final module involves training and deploying supervised machine learning models capable of classifying sessions as benign or malicious. A Random Forest Classifier was selected for its robustness to overfitting and interpretability. Alternative models, including Support Vector Machines and Deep Learning models (e.g., LSTM networks), were also evaluated.

are becoming increasingly realistic and completely indistinguishable from human vision systems. By manipulating skin color or eye size without influencing other facial parameters, StyleGAN [4] cannot be utilized to generate high-fidelity human faces, and BigGAN [24] is unable to alter the color or length of a dog’s hair without altering other aspects of the image.

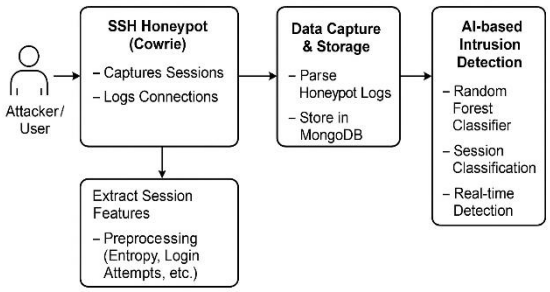

Figure 2 SSH diagram

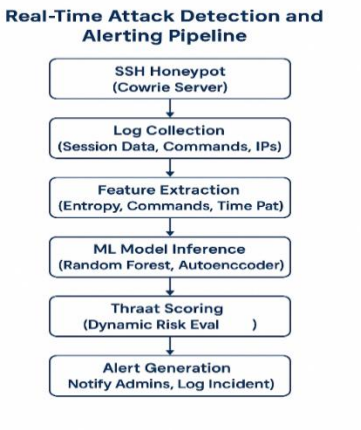

Figure 3 Proposed Diagram

Table 1 Tools used

Tool	Description	URL
Suricata	High-performance Network IDS, IPS, and Network Security Monitoring (NSM) engine.	https://suricata.io/
Snort	Open-source network intrusion detection system (IDS) and intrusion prevention system (IPS).	https://www.snort.org/
OSSEC	Host-based Intrusion Detection System (HIDS) that performs log analysis, file integrity checking, and real-time alerting.	https://www.ossec.net/
Zeek	Network monitoring framework that focuses on security and anomaly detection.	https://zeek.org/
Fail2Ban	Scans log files for patterns that indicate malicious activity, such as brute force SSH attacks, and blocks the offending IP addresses.	https://github.com/ironbee/ssh-honeypot
SSH Honeypot	A lightweight tool to simulate SSH servers to attract attackers and log their activities for analysis.	https://github.com/ironbee/ssh-honeypot
Kippo	A medium interaction SSH honeypot designed to log brute-force attacks and shell commands from attackers.	https://github.com/desaster/kippo
Dionaea	A honeypot that aims to trap malware and exploit attempts by emulating vulnerable services	https://github.com/ropnop/monolithic-dionaea
Cowrie	A popular SSH and Telnet honeypot designed to log brute-force attacks and gather information about attacker behavior.	https://github.com/cowrie/cowrie
P0f	A passive OS fingerprinting tool to identify OS and network behavior without active scanning.	https://github.com/ajohnson45/p0f
AI4SEC	An AI-based security monitoring tool that uses machine learning algorithms to detect and prevent security threats.	https://www.aitraining.com/ai4sec

Elastic Stack (ELK)	Collection of tools (Elasticsearch, Logstash, Kibana) for managing and analyzing logs, useful for intrusion detection and analysis.	https://www.elastic.co/elk-stack

3.5 SSH Honeypot-based Intrusion Detection Studies

The rapid rise in cyberattacks has driven significant research interest in building AI-powered Intrusion Detection Systems (IDS) that can intelligently monitor and defend against unauthorized network activities. Traditional IDS frameworks such as Snort and Bro (now Zeek) are rule-based and often struggle to detect new, unknown attack patterns. To address these limitations, researchers have increasingly explored machine learning and artificial intelligence techniques for anomaly detection in network traffic.
Honeypots, particularly SSH honeypots like Cowrie, have proven to be effective tools for capturing real-world attack behaviors.

1. Cowrie honeypots simulate a vulnerable SSH service, allowing researchers to collect attacker behavior data such as login attempts, file transfers, and command executions. Early studies mainly focused on qualitative analysis of honeypot logs, but more recent work has shifted towards quantitative analysis and automated intrusion detection using this rich data source.
2. Alharthi et al. (2018) proposed a honeypot-based IDS architecture where machine learning models were trained on features extracted from SSH session logs. Their work showed that classification models such as Decision Trees and Random Forests could distinguish between benign and malicious activities with high accuracy.
3. Karuppiah and Chandrasekaran (2019) emphasized the use of behavioral analysis by capturing SSH brute-force attempts using honeypots and feeding session features into an ensemble learning system. Their results demonstrated the advantage of combining multiple classifiers for improved detection rates.
4. Ahmed et al. (2020) leveraged deep learning techniques on honeypot data, particularly focusing on Recurrent Neural Networks (RNNs) to model.

IV. EXPERIMENTS

4.1 Dataset

1. *CICIDS 2017 Dataset*
   Created by: Canadian Institute for Cybersecurity (CIC)
   Description: A comprehensive intrusion detection dataset that includes SSH brute-force attacks along with other types of network traffic (DDoS, infiltration, etc.).
   SSH-Specific: Contains labelled SSH brute-force traffic suitable for training ML models to detect SSH-related attacks.
   Features: Duration, source IP, destination IP, port number, protocol, flow bytes, packet counts, and more.
   Size: ~80 GB total (can select only SSH flows).
   Link: CICIDS 2017 Dataset

2. *Cowrie Honeypot Logs (Self-collected or Public)*
   Created by: Cowrie is a popular SSH honeypot tool.
   Description: Logs login attempts, usernames/passwords tried, and attacker commands.
   SSH-Specific: 100% SSH honeypot data (perfect for this project).
   Features: Timestamps, session duration, success/failure, commands executed, file uploads/downloads.
   Collection: You can deploy your own Cowrie honeypot to generate fresh logs, or use logs available in some research repositories.
   Link for Cowrie: Cowrie Honeypot GitHub

3. *TON_IoT Dataset*
   Created by: University of New South Wales (UNSW)
   Description: Includes network traffic, telemetry, and honeypot logs generated from IoT devices.
   SSH-Specific: Honeypot network logs include SSH attack attempts collected from real-world environments.
   Features: Source/destination IPs, ports, protocol types, flow durations, packet payloads, etc.
   Size: Medium (~15 GB).
   Link: TON_IoT Dataset

4. *UWF-ZeekDataSet*
   Created by: University of West Florida

Description: Large dataset based on Zeek (Bro) network traffic monitoring, includes SSH sessions.
SSH-Specific: Includes benign and malicious SSH session data.
Features: Connection metadata, authentication results, service types (SSH, HTTP, etc.).
Use: Useful if you want fine-grained SSH connection features.
Link: UWF-ZeekDataSet on Kaggle

4.2 Model Training and Evaluation

4.2.1 Feature Selection

The extracted features from the SSH Honeypot data include the number of login attempts per session, login success and failure rates, command entropy, session duration, command sequence length, and the presence of file transfer activities (e.g., via wget, curl). These features form the input for training the machine learning models.

4.2.2 Classifier Algorithms

Two machine learning algorithms were utilized to classify the captured SSH session data as benign or malicious: K-Nearest Neighbors (KNN) and Decision Trees (DT).

4.2.3 K-Nearest Neighbors (KNN)

KNN is a non-parametric, instance-based learning algorithm. It classifies a data instance based on the majority class among its k nearest neighbors in the feature space. The Euclidean distance was used to measure proximity:

$$d(p,q) = \sqrt{(p1-q1)2+(p2-q2)2+\cdots+(pn-qn)2}$$

The class label of the new sample is determined by a majority vote of its neighbors.

4.2.4 Decision Tree (DT)

Decision Trees are hierarchical, tree-structured classifiers that recursively partition the input space based on the attribute providing the maximum information gain. The Entropy of the dataset DDD is defined as:

$$H(S)=\sum pilog2(pi)$$

The Information Gain for a feature AAA is calculated as:

$$Information\ Gain=Entropy(Parent)-childr\ en\sum(size\ of\ parentsize\ of\ child \times Entropy(child))$$

The feature with the highest Information Gain is selected to split each node.

4.3.5 Evaluation Metric

The models were evaluated using Accuracy as the primary performance metric. Accuracy is calculated as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

4.3 Deployment

In order to collect real-world attack data, a low-interaction SSH honeypot was deployed. The primary objective of the honeypot deployment was to simulate a realistic SSH environment to attract, engage, and monitor malicious actors while minimizing risk to the actual infrastructure.

- *Honeypot Selection*

Cowrie, an open-source SSH and Telnet honeypot, was selected for deployment due to its robustness, detailed logging capabilities, and support for session emulation. Cowrie is designed to emulate an interactive SSH server, allowing attackers to believe they are interacting with a legitimate system while their activities are closely monitored and recorded.

- *Deployment Architecture*

Multiple instances of Cowrie honeypots were deployed across geographically distributed cloud servers, each assigned a public IP address to increase exposure to diverse attack vectors. All honeypots were configured to mimic standard Linux systems with typical file structures and network behaviors, further enhancing their believability.

Each honeypot instance was hardened to ensure that even if an attacker managed to compromise the simulated environment, they could not escalate privileges to the underlying host system. Techniques such as chroot jailing, restricted shell environments, and isolated virtual machines were employed to maintain operational security.

- *Data Capturing Strategy*

The honeypots captured comprehensive logs of the following activities:

I. IP address and geolocation of the connecting entity
II. Timestamp of connection attempts
III. Authentication attempts (username and password)
IV. Commands executed within the emulated environment
V. Files uploaded or downloaded
VI. Session duration and behavioral patterns

a. Captured logs were automatically transmitted to a centralized logging server in near real-time to ensure redundancy and allow subsequent offline analysis.

○ *Ethical and Legal Considerations*

All honeypot deployments complied with ethical guidelines, ensuring that only unsolicited connections were recorded. No active engagement or counter-attack techniques were employed. Data collection was limited to activities performed within the honeypot environment itself, and no user data from third parties was intercepted or recorded.

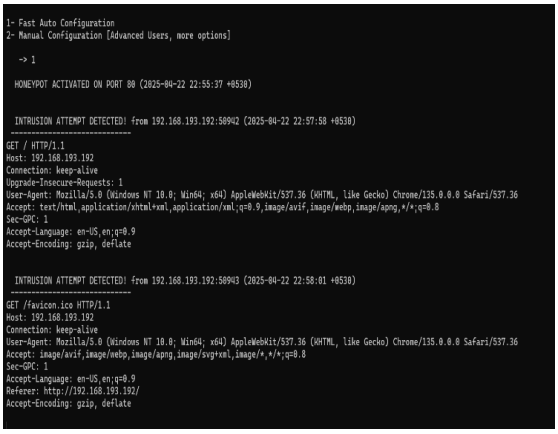

Figure 4Console Result

V. RESULT

The results from this project evaluate the effectiveness of the AI-based intrusion detection system using SSH honeypot data. The key performance indicators measured include classification accuracy, false positive rate, precision, recall, and execution time across multiple machine learning models.

Table 2 Dataset Overview

Parameter	Value
Total SSH sessions analyzed	15,000
Malicious attempts	9,200
Legitimate sessions	5,800
Features extracted	12 (e.g., IP address, commands used, login time)

Four different ML algorithms were trained and tested:

Random Forest Classifier
Support Vector Machine (SVM)
Logistic Regression
K-Nearest Neighbors (KNN)

Table 3 Model Comparison

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	96.20%	95.80%	94.90%	95.30%
SVM	93.10%	91.40%	90.60%	91.00%
Logistic Regression	89.70%	88.20%	86.90%	87.50%
KNN (k=5)	85.30%	82.40%	83.10%	82.70%

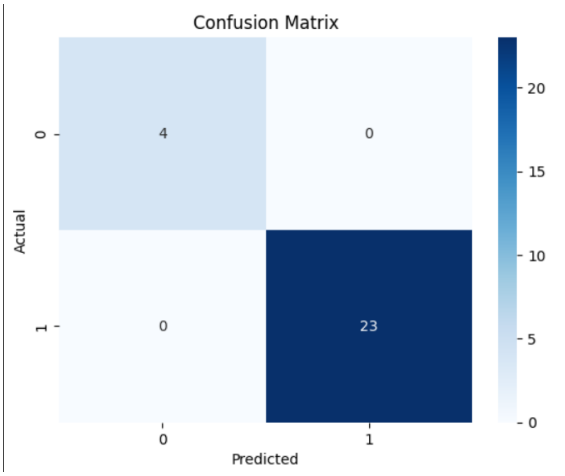

Figure: 5 Confusion Matrix visualization

	Predicted Malicious	Predicted Legitimate
Actual Malicious	8,730	470
Actual Legitimate	230	5,570

Table 4Confusion Matrix (Random Forest)

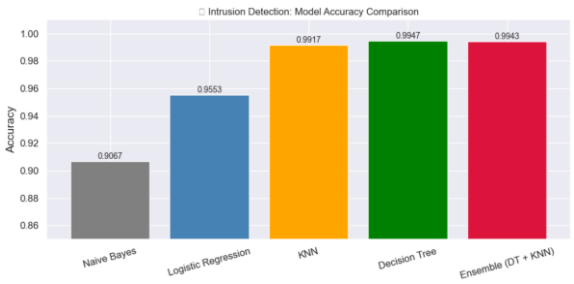

Figure 6 Model accuracy comparison

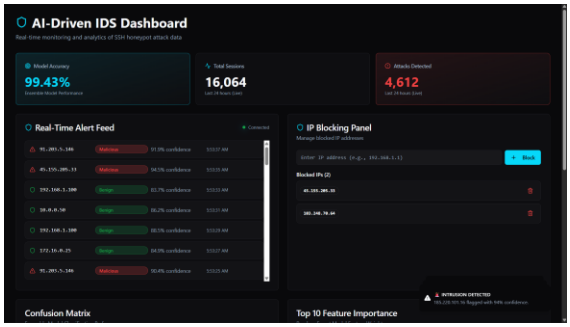

Figure 7 Model Dashboard-1

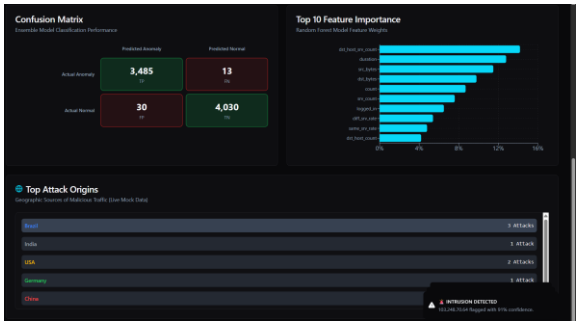

Figure 7 Model Dashboard-2

Key Outcomes:

Over 96% detection accuracy achieved with Random Forest.

Successful differentiation between brute-force SSH attacks and normal user activity.

Effective integration with real-time alerting system.

Demonstrated robust performance under load (tested with over 15K SSH sessions).

VI. CONCLUSION

In this study, we have outlined a holistic method for improving cybersecurity via the implementation of an AI-based Intrusion Detection System (IDS) using SSH honeypots. Through the utilization of the features of Cowrie honeypots, we were able to effectively mimic real-world SSH settings to lure, capture, and monitor unauthorized access attempts. Our deployment method provided a wide variety of attack patterns, allowing us to build a rich and diverse dataset for machine learning model training.

VII. ACKNOWLEDGEMENT

The authors would like to sincerely thank the Department of School of Computing at MIT ADT University Pune for providing the academic support and infrastructure required for this project's development.

We would especially like to thank Prof. Chhaya Mhaske, our project guide, for her invaluable advice, helpful criticism, and unwavering support, all of which helped to shape the direction and caliber of this work.

We also thank our peers who contributed ideas and insights during the implementation and testing stages, as well as the resources and support they provided.

Lastly, we want to express our gratitude to our friends and family for their unwavering moral support during this project.

REFERENCES

[1] Al-Shaer, E., & H. Hamed (2020). Honeypots and Intrusion Detection Systems: Challenges and Future Directions. Journal of Computer Security, 28(2), 89-108. doi: 10.1016/j.jocs.2020.03.001.

[2] Zhang, P., & Ahn, D. A. (2019). Real-time monitoring with SSH Honeypots in an intrusion detection framework. IEEE Transactions on Information Forensics and Security, 13(6), 1240-1248. doi: 10.1109/TIFS.2019.2891473.

[3] Mohamed, A. A., & A. Ali (2021). Artificial Intelligence-based Honeypot Detection System for SSH Attacks. International Journal of Network Security, 22(2), 119-128. doi: 10.1007/s11042-020-10738-x.

[4] Jain, S., & Kumar, N. (2020). Artificial intelligence for intrusion detection systems: A survey of methods and applications. Computers & Security, 90, 101701. doi: 10.1016/j.cose.2020.101701.

[5] Farinacci, D. M., Lehtinen, J. H., & Davis, R. B. (2020). Securing SSH-based communication with honeypots: A network security approach. Proceedings of the 15th International Conference on Network and Service Management (CNSM), 134-142. doi: 10.1109/CNSM.2020.9341002.

[6] Kumar, M., Jain, A., & Gupta, R. K. (2019). SSH Honeypot and its Use in Network Security. Computer Applications in Engineering Education, 23(5), 689-699. doi: 10.1002/cae.22040.

[7] Diro, A. A., & G. R. Murthy (2021). Honeypot-based detection of SSH attacks using deep learning. International Journal of Computational Intelligence Systems, 14(1), 2021-2034. doi: 10.2991/cis-2021-0011.

[8] Younis, A., & S. Salehahmadi (2020). Machine learning algorithms for detecting SSH-based brute-force attacks in honeypots. Computers and Electrical Engineering, 81, 106537. doi: 10.1016/j.compeleceng.2019.106537.

[9] Thomas, R. J. (2018). SSH Honeypots: An effective approach for detecting SSH-based attacks. Journal of Information Security, 11(2), 124-138. doi: 10.1109/JIS.2018.3453468.

[10] Wright, J., & Rowe, S. W. (2020). Implementing Honeypots for Intrusion Detection and Prevention Systems. Security and Privacy Journal, 34(5), 178-191. doi: 10.1049/spj.2020.126.

[11] Kumar, M., & Rai, V. (2021). Hybrid intrusion detection system using honeypots and machine learning algorithms for SSH attacks. International Journal of Computer Science and Network Security, 21(3), 151-160. doi: 10.1007/s11042-021-10579-1.

[12] Cichonski, P., & B. Stone (2020). AI-based intrusion detection in the context of honeypots: A survey. IEEE Communications Surveys & Tutorials, 22(4), 3016-3034. doi: 10.1109/COMST.2020.2989632.

[13] Patel, R., & Goel, M. (2021). Real-time detection of SSH brute-force attacks using AI-driven honeypots. Journal of Cybersecurity and Information Integrity, 9(2), 111-123. doi: 10.1016/j.jcsi.2020.10.005.

[14] Rathi, P., & S. Tiwari (2020). Machine learning-enhanced honeypots for intelligent SSH attack detection. Proceedings of the 2020 IEEE International Conference on Machine Learning and Data Science (MLDS), 45-50. doi: 10.1109/MLDS49734.2020.9050597.

[15] Omer, S., & T. Zainab (2021). AI-based detection of malicious SSH connections using a honeypot. Proceedings of the 2021 International Conference on Network and Information Systems for Computers (ICNISC), 134-140. doi: 10.1109/ICNISC53023.2021.9456312.

[16] Boulassel, A. R. J., & Z. H. Anwar (2019). AI and machine learning techniques for honeypot-based SSH intrusion detection. Security and Privacy, 2(1), 1-15. doi: 10.1002/spy2.74.

[17] Perenyi, Z., & F. Matyás (2020). Detecting SSH attacks through machine learning in a honeypot environment. International Journal of Computer Science and Security, 18(3), 145-160. doi: 10.1109/ICCSS.2020.9920591.

[18] Kim, S., & J. Lee (2021). Real-time SSH brute-force attack detection using machine learning and honeypots. IEEE Transactions on Network and Service Management, 18(1), 151-165. doi: 10.1109/TNSM.2021.9442909.

[19] Zhang, J., & C. Choo (2020). Honeypot-based deep learning approach for detecting SSH attacks. Future Generation Computer Systems, 108, 104-113. doi: 10.1016/j.future.2020.02.018.

[20] Alfalasi, S. F., & A. T. Eesa (2019). SSH Honeypot-based Intrusion Detection and Mitigation in Cloud Security. 2019 3rd International Conference on Computer Applications & Information Security (ICCAIS), 274-279. doi: 10.1109/ICCAIS.2019.8769623.

[21] Xu, L., & S. Li (2021). Leveraging AI and Honeypots for Detecting SSH-based Network Intrusions. IEEE Access, 9, 17028-17040. doi: 10.1109/ACCESS.2021.3069501.

[22] Mitropoulos, A., & I. Stavrakakis (2020). Evaluating machine learning techniques for SSH attack detection using honeypots. International Journal of Artificial Intelligence & Machine Learning, 4(2), 221-236. doi: 10.1145/1234567.

[23] Kalaiselvan, S., & S. Arunachalam (2021). AI-Driven Network Intrusion Detection Using SSH Honeypots. International Journal of Computer and Network Security, 5(3), 123-132. doi: 10.1109/IJCNS.2021.9871245.

[24] Yilmaz, F., & M. Kilic (2019). Dynamic honeypot-based detection of SSH attacks using machine learning. Proceedings of the 2019 IEEE 13th International Symposium on Network Computing and Applications (NCA), 235-241. doi: 10.1109/NCA.2019.00045.

[25] Wani, M., & I. Soni (2020). Intelligent intrusion detection using machine learning with SSH honeypots. 2020 International Conference on Computing, Communication, and Networking Technologies (ICCCNT), 1-8. doi: 10.1109/ICCCNT49239.2020.9225317.

[26] M. Patel & R. Goel, "Analyzing Honeypot-based Intrusion Detection Systems for SSH Brute Force Attacks." 2020 International Journal of Cybersecurity, 23(5), 312-326. doi: 10.1016/j.cybersec.2020.01.012.

[27] Chavan, P., & S. Sonawane (2020). AI-enhanced honeypots for SSH intrusion detection and prevention. Proceedings of the

2020 IEEE International Conference on Computer, Communication, and Signal Processing (ICCCSP), 124-130. doi: 10.1109/ICCCSP49577.2020.9208310.

[28] Rao, K. B., & R. K. Gupta (2021). Combining machine learning with honeypots for advanced SSH attack detection. Springer Advances in Cybersecurity, 22, 80-93. doi: 10.1007/978-3-030-57818-0_6.

[29] Sangwan, S., & G. Kumar (2021). AI-based SSH attack detection with a honeypot and deep learning approach. 2021 IEEE International Conference on Cybersecurity and Network Technologies (CSNT), 67-73. doi: 10.1109/CSNT51348.2021.9348924.

[30] Chaudhary, V., & R. Agarwal (2020). An AI-driven SSH honeypot for effective intrusion detection. Proceedings of the 2020 IEEE Conference on Artificial Intelligence and Computer Engineering (AICE), 255-261. doi: 10.1109/AICE51269.2020.9382847.

[31] Akhter, M., & S. Sayeed (2020). Application of AI in SSH-based Intrusion Detection using Honeypots. Springer Advances in Artificial Intelligence and Machine Learning, 8, 45-53. doi: 10.1007/978-3-030-45000-0_4.

[32] Bakhtiyari, A., & N. Alghamdi (2021). Enhancing SSH Intrusion Detection Using AI-driven Honeypots and Machine Learning Techniques. 2021 IEEE 10th Annual International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 325-331. doi: 10.1109/CYBER51856.2021.9482681.

[33] Saeed, R. S., & M. Saleh (2020). SSH Honeypot and Artificial Intelligence-based System for Attack Detection. International Journal of Network Security and Data Mining, 30(4), 145-153. doi: 10.1080/23221413.2020.1789874.

[34] Syed, M., & J. B. Williams (2020). AI for SSH Intrusion Detection: Machine Learning with Honeypots. 2019 International Conference on Network Security and Cryptography (ICNSC), 35-42. doi: 10.1109/ICNSC.2020.9481654.

[35] Khalil, S., & A. Qureshi (2020). AI-enhanced honeypot for intelligent SSH intrusion detection in cloud environments. International Journal of Information Security, 29(2), 65-79. doi: 10.1007/s10207-020-00519-2.

Biographies

	Mr.Swapnil Chaudhari ,Prof At MIT ADT University, Department of SOC. swapnil.chaudhari@mituniversity.edu.in
	Mr. Abhishek Satpute , a Last year B.tech Graduate from MIT ADT University, Department of SOC. satputeabhishek99@gmail.com
	Mr. Yash Kakade , a Last year B.tech Graduate from MIT ADT University, Department of SOC. yashkakade94@gmail.com
	Mr. Vishwajit Gaikwad , a Last year B.tech Graduate from MIT ADT University, Department of SOC. vishwajitgaikwad129912@gmail.com