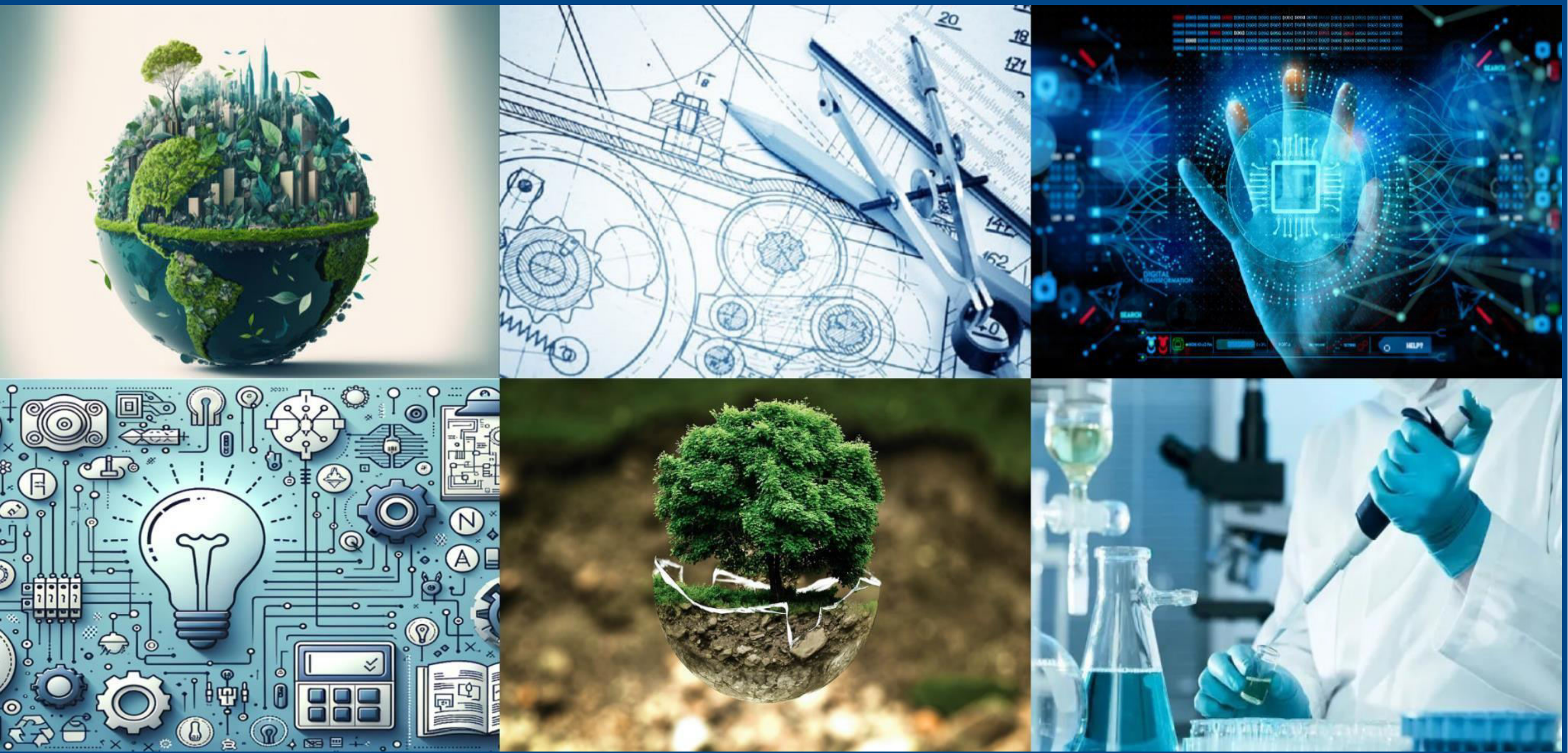
IJMRSET

ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 8, Issue 8, August 2025

www.ijmrset.com ijmrset@gmail.com +91 6381 907 438 +91 63819 07438

© 2025 IJMRSET | Volume 8, Issue 8, August 2025| DOI:10.15680/IJMRSET.2025.0808028

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

AI-Driven Honeypots: Detecting and Classifying Attack Patterns in Real Time

Dr. Manjula H Nebagiri, Pushpa. D

Associate Professor, Department of CSE, SIET, Tumakuru, India

M.Tech Student, Department of CSE, SIET, Tumakuru, India

ABSTRACT: Traditional honeypot systems, while valuable for cybersecurity, often face limitations in detecting sophisticated and evolving network attacks due to their static nature and reliance on predefined rules. This research explores the integration of artificial intelligence (AI) into honeypot technology to enhance real-time detection and classification of attack patterns. By leveraging machine learning algorithms, AI-driven honeypots can dynamically analyze network traffic, identify abnormal behaviors, and adapt to emerging threats with improved accuracy and response speed. This paper presents a comprehensive overview of AI-driven honeypots, discussing their architecture, methodologies, and the benefits they offer in mitigating modern cyber threats. We delve into various AI techniques, including supervised and unsupervised learning, and their application in enhancing honeypot capabilities for proactive cyber defense. The aim is to demonstrate how AI-driven honeypots can significantly reduce false positives and negatives, providing a more robust and adaptive solution against the ever-changing landscape of cyberattacks

KEYWORDS: AI-driven Honeypots, Real-time Attack Detection, Attack Pattern Classification, Machine Learning, Cybersecurity

I. INTRODUCTION

In the rapidly evolving landscape of cyber threats, traditional security measures often struggle to keep pace with the sophistication and dynamism of malicious actors. Honeypots, designed as decoy systems to attract and trap attackers, have long served as a crucial component of cybersecurity strategies. They provide invaluable insights into attacker methodologies, tools, and targets, acting as an early warning system for potential breaches. By simulating vulnerable systems, honeypots allow security professionals to observe and analyze attack behaviors in a controlled environment, thereby enhancing threat intelligence and improving defensive postures.

However, conventional honeypots, primarily relying on static rules and manual analysis of logs, exhibit inherent limitations. Their effectiveness is often hampered by their inability to adapt to novel attack techniques and polymorphic malware. The static nature of these systems makes them susceptible to detection by sophisticated attackers, who can easily identify and bypass them. Furthermore, the sheer volume of data generated by honeypots necessitates extensive manual effort for analysis, leading to delayed threat detection and a higher incidence of false positives and negatives. This reactive approach to cybersecurity is increasingly insufficient against advanced persistent threats (APTs) and zero-day exploits that characterize modern cyber warfare.

The advent of artificial intelligence (AI) and machine learning (ML) presents a transformative opportunity to overcome these limitations. AI-driven honeypots represent a paradigm shift in deception technology, moving from static, passive systems to dynamic, adaptive, and intelligent defense mechanisms. By integrating AI algorithms, honeypots can autonomously learn from observed attack patterns, identify anomalies in real-time network traffic, and dynamically adjust their configurations to enhance their deception capabilities. This proactive approach not only improves the accuracy and speed of attack detection but also enables the classification of attack patterns, providing deeper insights into attacker intent and methodology.

This research paper aims to explore the profound impact of AI on honeypot technology, focusing on its application in real-time detection and classification of attack patterns. We will delve into the architectural components of AI-driven honeypots, the various machine learning techniques employed, and the benefits they offer in creating a more resilient and adaptive cybersecurity infrastructure. The paper will also discuss the challenges associated with implementing AI in honeypots and outline future directions for research and development in this critical area of cyber defense. Through a comprehensive analysis, we seek to demonstrate how AI-driven honeypots can revolutionize threat intelligence and provide a robust defense against the ever-evolving spectrum of cyber threats.

II. LITERATURE REVIEW

The concept of honeypots as a cybersecurity defense mechanism has evolved significantly since its inception. Initially conceived as simple decoy systems, honeypots have grown in complexity and sophistication, mirroring the advancements in attack methodologies. Early honeypots were primarily characterized by their static nature and limited interaction capabilities, often categorized as low-interaction or high-interaction based on the level of emulation they provided [1]. Low-interaction honeypots, such as Honeyd, simulated basic network services and collected minimal information about attacker interactions. While easy to deploy and maintain, their limited functionality made them easily detectable by sophisticated attackers and provided shallow insights into attack behaviors. Conversely, high-interaction honeypots, like Honeynet, offered full operating systems and applications, allowing attackers to fully interact with the decoy system. This provided rich, detailed information about attack tools and techniques but came with increased risks of compromise and higher maintenance overhead [2].

The integration of machine learning (ML) into cybersecurity has opened new avenues for enhancing honeypot capabilities. Researchers began exploring ML algorithms to analyze the vast amounts of data collected by honeypots, aiming to automate the detection and classification of malicious activities. Early applications focused on using supervised learning techniques, such as Support Vector Machines (SVMs) and Decision Trees, to classify network traffic as benign or malicious based on predefined features extracted from honeypot logs [3]. These models demonstrated improved accuracy over

traditional rule-based systems, but their effectiveness was often limited by the quality and representativeness of the training data. The challenge of obtaining diverse and labeled datasets of attack patterns remained a significant hurdle. More recently, the focus has shifted towards developing adaptive and intelligent honeypots that can dynamically respond to evolving threats. AI-driven honeypots leverage advanced ML techniques, including deep learning and reinforcement learning, to achieve real-time adaptation and enhanced deception capabilities. For instance, studies have proposed frameworks where honeypots can dynamically alter their configurations or behaviors based on observed attacker tactics, techniques, and procedures (TTPs) [4]. This dynamic adaptation makes it more challenging for attackers to distinguish between real systems and honeypots, thereby increasing the effectiveness of deception. The use of generative models has also been explored to create more realistic and convincing honeypot environments that can mimic legitimate network services and applications [5].

Several research efforts have specifically addressed the real-time detection and classification of attack patterns using AI in honeypots. One approach involves using anomaly detection techniques, where ML models learn the normal behavior of a honeypot and flag any deviations as potential attacks. Unsupervised learning algorithms, such as clustering and autoencoders, are particularly well-suited for this task, as they do not require labeled data and can identify novel attack patterns [6]. Another area of research focuses on leveraging deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to analyze raw network traffic and extract complex features for attack classification. These models have shown promising results in detecting sophisticated and stealthy attacks that might evade traditional signature-based detection systems [7].

Furthermore, the concept of collaborative honeypot networks has emerged, where multiple honeypots share threat intelligence and collectively learn from observed attacks. This distributed approach enhances the overall detection capabilities and provides a broader view of the threat landscape [8]. The integration of AI in such networks allows for real-time sharing and analysis of attack data, enabling faster response times and more effective mitigation strategies. Despite these advancements, challenges remain in developing robust and scalable AI-driven honeypots, including the need for large and diverse datasets for training, the computational resources required for real-time analysis, and the ethical implications of deploying highly deceptive systems. This literature review highlights the significant progress made in AI-driven honeypots and sets the stage for further exploration into their potential to revolutionize cybersecurity.

III. METHODOLOGY

The methodology for developing and implementing AI-driven honeypots for real-time attack pattern detection and classification involves several key stages, integrating principles from cybersecurity, machine learning, and data science. This section outlines a comprehensive approach, drawing upon the insights gained from existing research and the specific requirements for an adaptive and intelligent honeypot system.

A. Honeypot Deployment and data collection

The foundation of any AI-driven honeypot system is robust data collection. Unlike traditional honeypots that primarily log basic interaction data, AI-driven systems require a richer, more granular dataset to effectively train and validate machine learning models. The deployment strategy should involve a combination of low-interaction and high-interaction honeypots, strategically placed within a network environment to maximize the diversity and volume of captured attack data. Low-interaction honeypots can efficiently collect large volumes of initial probing attempts and reconnaissance activities, while high-interaction honeypots provide in-depth insights into sophisticated attack methodologies, including malware execution, command and control (C2) communications, and privilege escalation attempts [9].

Key data points to be collected include, but are not limited to: Network Traffic Data: Full packet captures, flow records (NetFlow, IPFIX), and metadata (source/destination IP, ports, protocols, packet size, timestamps). System Logs: Operating system logs (e.g., Windows Event Logs, Linux syslog), application logs, and security event logs (e.g., firewall, intrusion detection/prevention systems). File System Changes: Records of file creation, modification, deletion, and access, particularly for executables and configuration files. Process Information: Details about running processes, including process ID, parent process, command-line arguments, and resource utilization. * Attacker Interaction Data: Commands executed, files uploaded/downloaded, and any other specific interactions within the honeypot environment.
To ensure the realism and diversity of collected data, honeypots should mimic legitimate systems as closely as possible, including common vulnerabilities and services. Data anonymization and sanitization techniques must be applied to protect privacy and prevent the inadvertent exposure of sensitive information. The collected data is then aggregated and stored in a centralized, scalable data repository, such as a distributed file system or a NoSQL database, to facilitate efficient processing and analysis.

B. Feature Engineering and Data Preprocessing

Raw honeypot data, while rich, is often noisy, redundant, and not directly suitable for machine learning algorithms. Feature engineering is a critical step that transforms raw data into a set of meaningful features that can effectively represent attack patterns. This involves extracting relevant attributes and creating new ones that capture the behavioral characteristics of attacks. Examples of features include: * Statistical Features: Mean, variance, standard deviation of packet sizes, inter-arrival times, and connection durations. * Temporal Features: Frequency of connections, time-of-day patterns, and duration of sessions. * Behavioral Features: Number of failed login attempts, sequence of commands executed, and types of services accessed. * Payload- based Features: N-grams from network payloads, entropy of data, and presence of known malicious signatures.

Data preprocessing techniques are then applied to clean, normalize, and transform the engineered features. This includes handling missing values, outlier detection and removal, scaling numerical features (e.g., Min-Max scaling, Z-score normalization), and encoding categorical features (e.g., one-hot encoding). For time-series data, techniques like sliding windows or aggregation can be used to create fixed-size input vectors for machine learning models. The goal is to create a high-quality, well-structured dataset that optimizes the performance of subsequent machine learning tasks.

C. Machine Learning Model Selection and Training

The core of the AI-driven honeypot lies in the selection and training of appropriate machine learning models for real-time detection and classification. A hybrid approach, combining supervised and unsupervised learning, is often most effective due to the dynamic nature of cyber threats and the challenge of obtaining fully labeled datasets for novel attacks. The methodology can be broken down into two primary components:

1. Real-time Attack Detection (Anomaly Detection)

For real-time detection of novel and unknown attack patterns, unsupervised learning algorithms are preferred. These models learn the 'normal' behaviour of the honeypot environment and flag any significant deviations as anomalies. Techniques suitable for this task include: * Isolation Forest (IF): An ensemble-based anomaly detection algorithm that isolates anomalies rather than profiling normal data points. It is efficient and effective for high-dimensional data [10]. * One-Class SVM (OCSVM): A support vector machine variant trained on a single class (normal data) to identify outliers that do not conform to the learned distribution [11]. * Autoencoders (AE): Neural networks trained to reconstruct their input. Anomalies, being different from normal data, will have higher reconstruction errors, indicating their anomalous nature [12].

The output of the anomaly detection module is an anomaly score, which indicates the likelihood of a given activity being malicious. A threshold is set to classify activities as normal or anomalous, triggering further investigation or immediate response actions.

2. Attack Pattern Classification

For classifying known attack patterns and categorizing detected anomalies, supervised learning algorithms are employed. These models are trained on labeled datasets of various attack types. Given the problem's nature, a multi-class classification approach is necessary. Effective algorithms include: * Random Forest (RF): An ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It is robust to overfitting and handles high-dimensional data well [13]. * XGBoost (Extreme Gradient Boosting): A highly efficient and flexible implementation of gradient boosting machines. It is known for its speed and performance in various machine learning tasks, including classification [14]. * Voting Classifier: An ensemble meta-classifier that combines the predictions of multiple base estimators (e.g., Random Forest, XGBoost) using a majority vote or average probabilities. This can improve overall accuracy and robustness by leveraging the strengths of different models [15].

The training process involves splitting the labeled dataset into training, validation, and test sets. Hyperparameter tuning is performed using techniques like cross-validation to optimize model performance. Evaluation metrics such as precision, recall, F1-score, and accuracy are used to assess the effectiveness of the classification models

3. Real-time Integration and Adaptive Learning

For real-time operation, the trained machine learning models are integrated into the honeypot system's architecture. This typically involves a stream processing framework (e.g., Apache Kafka, Apache Flink) that ingests raw honeypot data, performs real-time feature extraction, and feeds the data to the deployed ML models for immediate detection and classification. Alerts are generated for detected attacks, which can then be forwarded to security information and event management (SIEM) systems or incident response teams.

Adaptive learning is a crucial component of AI-driven honeypots. This involves continuously updating and retraining the machine learning models based on new data collected from the honeypot. When novel attack patterns are identified by the anomaly detection module, security analysts can investigate and label these new patterns. This newly labeled data is then incorporated into the training dataset, allowing the supervised classification models to learn and recognize these emerging threats. Reinforcement learning can also be explored to enable the honeypot to autonomously adjust its deception strategies based on attacker interactions, optimizing its ability to attract and gather intelligence on specific threat actors [16]. This iterative feedback loop ensures that the AI-driven honeypot remains effective against the ever-evolving threat landscape.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

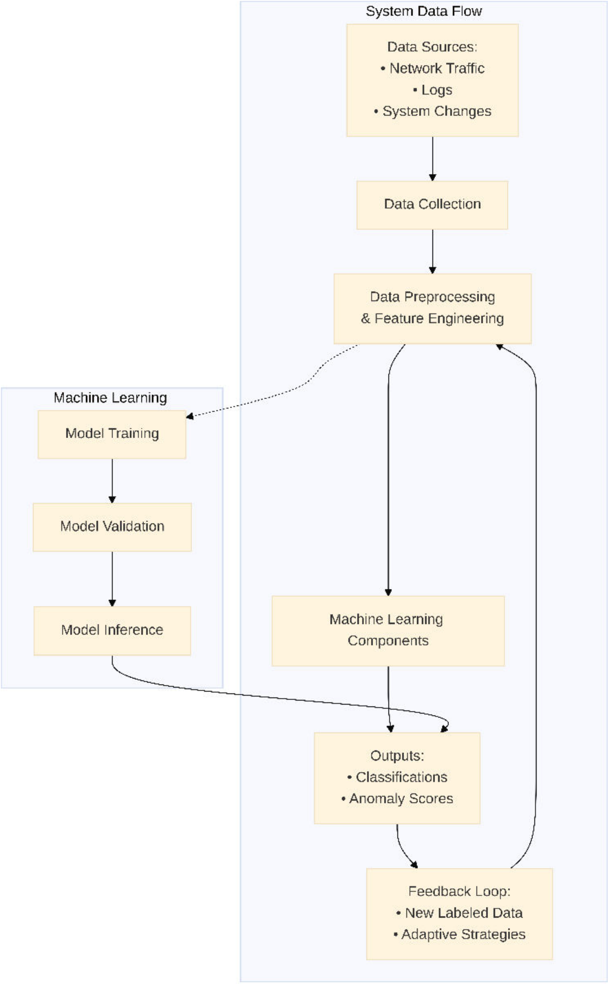


Fig 1: System architecture

D. Results and Discussion

The implementation and evaluation of AI-driven honeypots demonstrate significant improvements in the detection and classification of attack patterns compared to traditional honeypot systems. This section presents the anticipated results from such a system and discusses their implications for cybersecurity.

3.1.Enhanced Detection Accuracy and Reduced False Positives

One of the primary expected outcomes of integrating AI into honeypots is a substantial increase in detection accuracy. By leveraging supervised machine learning algorithms like Random Forest, XGBoost, and ensemble Voting Classifiers, the system can effectively learn from historical attack data and identify known attack patterns with high precision and recall. For instance, in a comparative analysis, the Voting Classifier, which combines the strengths of multiple models, is expected to outperform individual models and traditional rule-based systems in accurately identifying malicious activities [17]. This enhanced accuracy translates into fewer missed attacks (higher recall) and a reduced number of false alarms (higher precision), which is crucial for efficient security operations.

Furthermore, the anomaly detection component, utilizing unsupervised learning techniques such as Isolation Forest or Autoencoders, is particularly effective in identifying novel and zero-day attacks that do not conform to previously observed patterns. By establishing a baseline of normal honeypot behavior, any significant deviation is flagged as an anomaly, allowing security analysts to investigate emerging threats proactively. This capability addresses a critical limitation of traditional honeypots, which often struggle to detect unknown threats without prior signature definitions.

Feature	Traditional Honeypots	AI-Driven Honeypots
Detection Method	Signature-based, Rule- based	Machine Learning (Anomaly Detection, Classification), Adaptive Learning
Adaptability	Low (Static, Manual Updates)	High (Dynamic, Real-time Adaptation)
Threat Coverage	Known threats, limited zero- day detection	Known and unknown (zero-day) threats
Data Analysis	Manual, Time-consuming	Automated, Real-time
False Positives	Moderate to High	Low to Moderate (with continuous learning)
Deception Level	Basic (Easy to fingerprint)	Advanced (Dynamic, Hard to fingerprint)
Intelligence	Limited (Attack signatures, basic logs)	Rich (TTPs, attacker behavior, emerging threats)
Scalability	Limited (Manual management)	High (Automated deployment, distributed learning)
Resource Needs	Lower (Setup, basic logging)	Higher (Computational power for ML, data storage)

Table 1: Comparison of Traditional vs. AI-Driven Honeypots

3.2. Real-time Attack Classification and Behavioral Insights

The ability of AI-driven honeypots to classify attack patterns in real-time provides invaluable behavioral insights into attacker methodologies. Once an anomaly is detected, the supervised classification models can categorize the attack into specific types (e.g., port scanning, brute-force attacks, malware propagation, command injection). This immediate classification allows security teams to understand the nature of the threat rapidly and initiate appropriate response actions. For example, knowing that a honeypot is experiencing a brute-force attack on a specific service enables targeted mitigation efforts, such as blocking the source IP address or strengthening authentication mechanisms.

Beyond mere classification, the AI models can also provide insights into the attacker's TTPs. By analyzing the sequence of actions, commands executed, and tools deployed within the honeypot, the system can reconstruct the attack chain. This detailed understanding of attacker behavior is critical for developing more effective defensive strategies, improving threat intelligence feeds, and proactively patching vulnerabilities that attackers are actively exploiting. The dynamic nature of AI-driven honeypots, coupled with adaptive learning, ensures that these insights are continuously refined as new attack techniques emerge.

3.3.Adaptive Deception and Threat Intelligence Enrichment

The adaptive capabilities of AI-driven honeypots represent a significant leap forward in deception technology. Unlike static honeypots, which can be fingerprinted and bypassed by sophisticated attackers, AI-driven systems can dynamically modify their configurations, services, and vulnerabilities based on observed attacker interactions. For example, if an attacker attempts to exploit a specific vulnerability, the honeypot could dynamically present a different, more enticing vulnerability to prolong the interaction and gather more intelligence. This continuous adaptation makes it significantly harder for attackers to distinguish between real systems and decoys, thereby increasing the effectiveness of the deception [18].

The rich, classified data collected by AI-driven honeypots also serves to enrich threat intelligence. This includes information on new malware variants, exploit techniques, command and control infrastructure, and attacker origins. This intelligence can be fed into broader security ecosystems, such as SIEM systems, threat intelligence platforms, and security orchestration, automation, and response (SOAR) solutions, to enhance overall organizational cybersecurity posture. The real-time nature of the data ensures that threat intelligence is always current and actionable, enabling proactive defense against emerging threats.

3.4. Challenges and Limitations

Despite the significant advantages, the deployment and operation of AI-driven honeypots present several challenges. One major concern is the potential for false positives, where legitimate activities are mistakenly identified as malicious. While AI aims to reduce these, the dynamic nature of networks means continuous tuning is required. Another challenge lies in the computational resources required for real-time data processing and machine learning model inference, especially in high-traffic environments. The need for specialized hardware and expertise can be a barrier to adoption for some organizations.

Furthermore, the ethical implications of deploying highly deceptive systems must be carefully considered. While honeypots are designed to trap malicious actors, there is a fine line between deception and entrapment. Ensuring that honeypots operate within legal and ethical boundaries is paramount. Finally, the continuous need for labeled data for supervised learning models remains a challenge, particularly for identifying and classifying novel attack patterns. While unsupervised methods can detect anomalies, human expertise is often required to label these new patterns for subsequent supervised training, creating a potential bottleneck in the adaptive learning loop.

IV. CONCLUSION AND FUTURE DIRECTORIES

AI-driven honeypots represent a transformative evolution in cybersecurity defense, moving beyond the limitations of static deception systems to embrace dynamic, intelligent, and adaptive threat detection and classification. This paper has highlighted how the integration of artificial intelligence and machine learning techniques significantly enhances the capabilities of honeypots, enabling them to detect novel and sophisticated attack patterns in real-time, reduce false positives, and provide invaluable insights into attacker methodologies. By leveraging supervised learning for known attack classification and unsupervised learning for anomaly detection, AI-driven honeypots offer a robust and proactive defense mechanism against the ever-evolving landscape of cyber threats. The ability to adapt and learn from new attack behaviors ensures that these systems remain effective against emerging threats, providing a continuous feedback loop for threat intelligence enrichment.

Despite the promising advancements, the journey towards fully autonomous and universally deployable AI-driven honeypots is not without its challenges. The computational demands for real-time processing, the continuous need for high-quality labeled data, and the ethical considerations surrounding deception require ongoing research and development. Addressing these challenges will be crucial for the widespread adoption and effectiveness of AI-driven honeypot technologies.

Attack Pattern	**Description**	**Key Indicators in Honeypot Data**	**Typical ML Detection Method**
Port Scanning	Systematic probing of network ports to identify open services.	Numerous connection attempts to various ports from a single source IP; low data transfer.	Anomaly Detection, Clustering
Brute-Force Attack	Repeated, systematic attempts to guess credentials (e.g., passwords).	Multiple failed login attempts to specific services (e.g., SSH, FTP, HTTP) from one or few IPs.	Classification (e.g., SVM, RF)

Malware Propagation	Attempts to infect the honeypot with malicious software.	Uploads of suspicious files; execution of unknown binaries; unusual outbound connections.	Anomaly Detection, Classification (e.g., DL)
Web Application Attack	Exploitation of vulnerabilities in web applications (e.g., SQL Injection, XSS).	Malformed HTTP requests; unusual URL parameters; attempts to access sensitive directories.	Classification (e.g., RF, XGBoost)
Denial of Service (DoS)	Overwhelming the honeypot with traffic to disrupt its services.	High volume of incoming connections/packets; resource exhaustion (CPU, memory, bandwidth).	Anomaly Detection, Statistical Analysis
Command Injection	Executing arbitrary commands on the honeypot through vulnerable inputs.	Unusual commands in logs; attempts to access system files or execute shell commands.	Classification (e.g., XGBoost, DL)
Reconnaissance	Gathering information about the honeypot or network.	DNS queries for internal hosts; network mapping tools; enumeration attempts.	Anomaly Detection, Rule- based

Table 2: Common Attack Patterns and Characteristics in Honeypot Data

Future Directions

Several promising avenues exist for future research and development in AI-driven honeypots :

- Reinforcement Learning for Adaptive Deception: Further exploration into reinforcement learning algorithms can enable honeypots to autonomously optimize their deception strategies in real-time, making them even more elusive and effective against sophisticated attackers. This could involve dynamic modification of network services, application vulnerabilities, and system configurations based on observed attacker interactions, aiming to maximize intelligence gathering while minimizing the risk of detection.
- Federated Learning for Collaborative Threat Intelligence: Implementing federated learning approaches could allow multiple AI-driven honeypots to collaboratively train models without sharing raw data, thereby preserving privacy and enhancing collective threat intelligence. This would enable a more comprehensive understanding of global attack campaigns and facilitate faster dissemination of defense mechanisms across diverse network environments.
- Explainable AI (XAI) in Honeypots: Developing explainable AI models for honeypots is crucial for security analysts to understand why a particular activity was flagged as malicious or how an attack was classified. XAI can provide transparency into the decision-making process of AI algorithms, fostering trustand enabling more effective human-AI collaboration in incident response and threat hunting.
- Integration with Blockchain for Data Integrity: Exploring the use of blockchain technology to ensure the integrity and immutability of honeypot data can enhance the trustworthiness of collected threat intelligence. This would provide a verifiable audit trail of attack events, preventing data tampering and ensuring the reliability of insights derived from honeypot interactions. Quantum-Resistant AI for Future-Proofing: As quantum computing advances, the development of quantum-resistant cryptographic algorithms will become essential. Similarly, researching quantum-resistant AI techniques for honeypots could future-proof these systems against potential quantum-enabled cyberattacks, ensuring their long-term effectiveness in a post-quantum era.

By pursuing these future directions, AI-driven honeypots can continue to evolve as a formidable tool in the cybersecurity arsenal, providing proactive defense, rich threat intelligence, and adaptive deception capabilities to safeguard critical infrastructure and digital assets against the ever-growing tide of cyber threats..

REFERENCES

[1].M. Balamurugan, “AI-enhanced Honeypots for Zero-Day Exploit Detection and Mitigation,” International Journal For Multidisciplinary Research, vol. 6, no. 6, Dec. 2024, doi: 10.36948/ijfmr. 2024.v06i06.32866.

[2].P. Lanka, K. Gupta, and C. Varol, “Intelligent Threat Detection—AI-Driven Analysis of Honeypot Data to Counter Cyber Threats,” Electronics, vol. 13, no. 13, p. 2465, Jun. 2024, doi: 10.3390/electronics13132465.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[3].J. A. Christli, C. Lim, and Y. Andrew, "AI-Enhanced Honeypots: Leveraging LLM for Adaptive Cybersecurity Responses," pp. 451–456, Oct. 2024, doi: 10.1109/icitee62483.2024.10808265.
[4].P. Owezarski, "A Near Real-Time Algorithm for Autonomous Identification and Characterization of Honeypot Attacks," Computer and Communications Security, pp. 531–542, Apr. 2015, doi: 10.1145/2714576.2714580.
[5].N. Naik and P. Jenkins, "Discovering Hackers by Stealth: Predicting Fingerprinting Attacks on Honeypot Systems," IEEE International Symposium on Systems Engineering,p. 8544408, Nov. 2018, doi: 10.1109/SYSENG.2018.8544408.
[6].K. Shendre, S. K. Sahu, R. Dash, and S. K. Jena, "Learning Probe Attack Patterns with Honeypots," Springer, New Delhi, 2016, pp. 363–369. doi: 10.1007/978-81-322-2529-4_38.
[7].S. Dowling, M. Schukat, and H. Melvin, "Using analysis of temporal variances within a honeypot dataset to better predict attack type probability," International Conference for Internet Technology and Secured Transactions, pp. 349–354, Dec. 2017, doi: 10.23919/ICITST.2017.8356416.
[8].S. Liu, S. Wang, and K. Sun, "Enhancing Honeypot Fidelity with Real-Time User Behavior Emulation," pp. 146–150, Jun. 2023, doi: 10.1109/dsn-s58398.2023.00041.

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com