Sneha Mishra al. International Journal of Recent Research Aspects ISSN: 2349-7688, Special Issue: International Conference on Intelligent Systems for Complex, Chaotic, & Connected Environments Sept 2025, pp. 25-31

Adaptive Honeypots for HTTPS

[1]Sneha Mishra, [2]Ishika Dhiman, [3]Kalidindi Sowmya

Department of Computer Science and Engineering, Sharda School of Engineering & Technology-Greater Noida, India

snehamishr008@gmail.com1, ishikadhiman123@gmail.com2, sowmyanagaraju0@gmail.com3

ABSTRACT- **The increase in sophistication of cyberattacks against encrypted web traffic has uncovered the limitations of traditional honeypots, which are static and easily discoverable. In light of this limitation, this research presents an Adaptive HTTPS Honeypot that combines dynamic deception techniques with machine learning-based attack classification. The framework consists of a simulated HTTPS server to engage with adversaries while a traffic collection and feature extraction module gathers both metadata and encrypted communication patterns that are subsequently analysed using supervised learning algorithms to identify brute-force, reconnaissance scans, and exploitation attacks in real time. An adaptive response engine modifies the honeypot's hooking behaviour, including SSL/TLS certificates, headers, open services, or simulated vulnerabilities, to maintain a longer contact time and make fingerprints harder. A central ELK-based the dashboard gives an analyst the ability to constantly monitor, visualize, and assess forensic data to make inform decisions. The alternative of dynamic transactions and scalable systems, provide resistance to the evolving threats facing HTTPS. The adaptive method is applicable to enterprise networks, LANs, and IoT devices. In summary, the contribution to design intelligent, adjusting, honeypots capable of detection and adaption modern methods used by adversaries for exploitation.**

Keywords— *Adaptive Honeypot, HTTPS Security, Machine Learning, Cyber Deception, Threat Intelligence*

I. INTRODUCTION

Cyber threats encompass harmful activities aimed at compromising the confidentiality, integrity, and availability of information systems. These activities may involve multiple actors, such as hackers working alone or in groups, criminal organizations, hacktivists, or sponsored attackers. Such actors take advantage of vulnerabilities to steal sensitive data, disrupt systems and/or cause financial harm. Because these attacks continue to evolve in their approach and technique, it is difficult to remain resilient when addressing these evolving threats. Governments can encounter threatening consequences to their society and politics, while the private sectors including finance, healthcare, and retail can find themselves targeted with ransomware attacks, phishing attacks, and theft of consumer data. As organizations increasingly depend on their digital systems and the services that come with them, all of that can be more secure if supported by a layered approach to security, employee online safety training, monitoring, and coordinated incident response. Communication and collaboration between public/private stakeholders that support critical infrastructure creates resilience through shared intelligence and coordinated action.

Honeypots provide an active defense which is beyond what current security tools will provide. A honeypot is a decoy which replicates an asset, with the intention of luring an attacker into a controlled environment where they can see the behavior of the attack, identify vulnerabilities, and gather threat intelligence – all while leaving real live systems unaffected. Honeypots will provide insight into the Tactics, Techniques, and Procedures (TTPs) of the attacker –

improving defenses against future threat events against the organization. Honeypots may be useful for monitoring unauthorized access to a public infrastructure or the enterprise, and they can also work in conjunction with firewalls, intrusion detection systems and continuous monitoring. Honeypots and other collected data can contribute to feed machine learning models for predictive threat detection and proactive security.

Traditional low-interaction honeypots provide services, or open ports to trick attackers as they pose a very low risk to the network. They passively collect intrusion attempts, IP addresses, payloads and methods of attack. However, due to their limitations, they provide little or no understanding of the attacker's behavior. Low interaction honeypots are easy to implement and very well suited for lower resource organizations or as initial deception layers in larger organizational networks. They can be used to observe automated attacks such as brute-force logins and being probed or scanned other security activity. However, skilled attackers may identify these honeypots and they will lose effectiveness, particularly against advanced persistent threats (APTs). Although less effective, low interaction honeypots can still provide some level of early detection and awareness of threats.

Honeypots that rely on machine learning (ML) analyze live traffic, commands, and payloads that help to uncover covert techniques used by attackers. Supervised models will classify known threats, unsupervised models will detect anomalies, and reinforcement learning will adapt to the behavior of attackers. The ML systems can automate threat intelligence, improve the precision of detection, and reduce workloads on human analysts. Some challenges for ML honeypots are dependence on quality training data, vulnerability to adversarial ML attacks, and the computational overhead introduced. However, ML honeypots can more effectively improve defensive capabilities through real-time classifications of traffic inputs, dynamic deception of malicious actors, and prioritization of alerts escalated by their potential for risk.

This research suggests an Adaptive Honeypot that merges dynamic deception with ML classification. The honeypot collects detailed information on network traffic and interactions using Cowrie and Scapy to collect, the adaptive engine changes ports, banners, file structures, and fake vulnerabilities, ML then classifies the interactions into brute-force, reconnaissance, privilege escalation, and exploit-type attacks. The information is then output via ELK stack visualization for real-time situational awareness, historical analytical capture, and actionable intelligence. The framework is identified as a scalable, intelligent, and self-adapting defense capable of extending honeypot abilities and enhancing resilience across government, enterprise, and cloud environments.

II. LITERATURE REVIEW

Honeypots are decoy systems created to lure attackers so that researchers are able to gather intelligence around emerging threats without putting actual systems at risk. Honeypots imitate vulnerable services and applications, so they can provide research insight into attacker behavior within a controlled environment. Successful, traditional honeypots can identify unauthorized systems access and malware; however, static honeypots typically lack effectiveness against adaptable attacks or AI-based attacks, depending instead on mounted defenses against scans to essentially obsolete malicious behavior. There are trade-offs with specific types of honeypots - low interaction honeypots have lower resource requirements, but they do not provide much insight for researchers; high interaction honeypots are real decoy systems that create the perception of a vulnerable system but require many operational resources and security measures for the honeypot.

Modern cyber threats, such as advanced persistent threats (APTs), polymorphic malware and zero-day exploits, are increasingly sophisticated and able to defeat static cybersecurity defenses. These kinds of attacks demand more than static approaches, which is the premise behind the establishment of honeypots that can modify their systems, network responses, and interaction levels dynamically in response to what the honeypot determines the attacker is doing. Adaptive honeypots have, when applied with approaches based on machine learning (ML) or artificial intelligence (AI), has made possible continuous profiling of attacker's behavior in real time™, detecting anomalies, classifying the types of attacks, and eventually predicting next actions. They can also use multiple types of deception by opening fake services, rotating ports, or simulate vulnerabilities while pursuing adversaries or misdirecting their interest. Continued engagement can both distract the adversary from critical organizational assets while generating cyber threat intelligence (CTI) for teams responsible for security that can help enhance the effectiveness of Speed of Incident Response, Threat Analysis and Proactive Defense actions. In summary, adaptive honeypots offer a critical cyber defense capability in building resilience to an ever-expanding base of complex and evolving threats.

The integration of continuous updates into a honeypot environment can come from utilizing threat intelligence feeds, both IoCs and adversarial tactical elements of both. The deception framework of honeypots, or orchestrated deception frameworks around honeypots, such as Cowrie, Dionaea, and Honeyd, has been developed with fully orchestrated deception frameworks to leverage the emulation of realistic services, dynamic user interactions and automated intelligence capture to augment awareness and defend an optimum cyber environment.

Adaptive honeypots use different techniques: SDN-based systems are adaptable at the network level, but rely on an additional underlying layer. SDN-based systems can introduce latency, or a single point of failure. ML-based systems can make use of behavior intelligence, but require increased computational resource and data sets, with reduced risk from adversarial attacks. A hybrid system based on SDN and ML approaches is multi-layered, adaptable, and has data collection, detection reduction capability, and time-on-target capability, but still require intensive computing to be effective.

Deployments face challenges with scalability in extensive networks, managing large volumes of interaction data, changes to how attackers evade honeypots, and complexity of integration. Other considerations include ethical, legal, and privacy factors such as data ownership and GDPR compliance which adds to the complexity of operations. All of this highlights the need for lightweight, automated, and resilient adaptive honeypot frameworks that effectively operate in modern adversarial environments.
Although there has been progress made, modern adaptive honeypots still have challenges such as semi-automated adjustment, lagging real-time adjustments, limited coverage of the threat landscape, reliance on high-quality data sets, and usability concerns. The goal of this research is to inform a gap in the literature by developing a self-evolving adaptive honeypot that employs dynamic deception, real-time machine learning-based threat classification, and automated behavioral adaptation to provide a scalable and intelligent cyber defense mechanism.

III. METHODOLOGY

This study adopts a modular, adaptive method to deploy a honeypot modelled on HTTP traffic to capture event-level activity on a local area network (LAN). Each implementation goes through a process of analyzing every aspect of attacker-based behavior, utilizing machine learning algorithms in the detection of attacks, and adapting in real time to the response procedure. The overall methodology is defined in five sections: Traffic Collection, Data Preprocessing & Feature Engineering, Machine Learning Algorithm Build, the Adaptive Interaction Layer, and System Deployment and Evaluation.

A. Traffic Collection

In the first phase, both controlled LAN environments and synthetic attack scenarios collect HTTP request data. The honeypot operates as a decoy web server that hosts fake but realistic web applications designed to emulate common

vulnerabilities to attract potential attackers. Every interaction, including HTTP methods, headers, payloads, and client metadata, is captured in real time. In order to further extend the dataset and improve the model's generalizability, extra data from publicly available datasets like CICIDS and UNSW-NB15 are included. The traffic collected from both environments includes a diverse array of both legitimate and malicious HTTP requests that are used to train and validate the detection model.

B. Data Preprocessing and Feature Engineering

After gathering the traffic data, it is processed and cleaned to prepare it for machine learning. This includes removing duplicates and incomplete packets, and deriving relevant features from the HTTP session. Engineered features include the following: the request frequency, the URL path length, delay in response, payload size, and entropy. For supervised learning, the data set is labeled with a combination of automated rule detection (using IDS tools, such as Suricata or Snort) and manually monitoring the packets. Each request is labeled as benign or malicious to create a sufficiently robust labeled dataset to train the classification model.

C. Machine Learning Model Construction

The labeled dataset is then used to develop a supervised machine learning model that can classify HTTP traffic and identify attack types. In other words, several algorithms (e.g., decision trees, random forests, and neural networks) are evaluated to select the best model(s) created in terms of accuracy, yet factoring in the computational cost as well. The dataset is separated into training and testing data (sometimes 80:20 is used), and model performance is validated using cross-validation methods. Finally, hyperparameter tuning is performed to improve the model detection accuracy, and the final model outputs a classification label and a probability score on the confidence level of malicious activity (SQL injection, XSS injection, brute force, etc.).

D. Adaptive Interaction Layer

The adaptive interaction engine is at the center of the system, reacting to the output from the machine learning model in order to modify the honeypot behavior. When it detects low-risk or exploratory conduct, the system will continue to project inaccurate but plausible content in order to gather more actionable intelligence. When an attack is high confidence, the system may deliberately impose artificial delays, present misleading responses, or provoke the behavior response to generate confusion and a deterrent effect.

Adaptive logic applies previous interaction feedback with the intention of maturing over time. Upon observing attack patterns, the honeypot model updates the behavior profile which keeps the system robust and unpredictable towards constantly evolving attack tactics.

E. System Architecture

The Adaptive HTTPS Honeypot has several interworking components that dynamically interact to attract attackers and gather intelligence. Its system architecture has the following main modules:

- Attacker Machine: Sends malicious HTTPS requests to the Honeypot.
- Honeypot Web Server: Simulates realistic HTTPS services (login, admin, APIs) to attract attackers.
- Request Logger: Collects detailed information about arriving requests (e.g., IP, payload).
- Feature Extractor: Processes the request data, creating features such as request frequency and payload size.
- ML Detection Engine: Classifies incoming traffic as either benign or malicious using machine learning algorithms.
- Adaptive Decision Engine: Dynamically alters the Honeypot's actions (e.g., rotating ports, changing certificates) based on the classification of the attack.
- Response Profile Manager: Switches between fake service profiles to negatively impact attackers.
- Storage and Dashboard: Deploys the ELK stack to provide real-time monitorization, logging, and visualization of attacker behavior.

Figure 1 illustrates the data flow between these modules and how each one cooperatively works to detect, engage, and react to malicious activity in real-time.

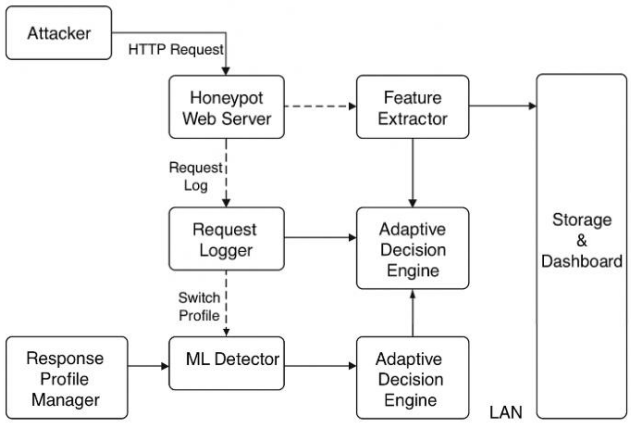

Fig. 1. System architecture of the Adaptive HTTPS Honeypot

F. Deployment and Performance Evaluation

The last stage consists of deploying the honeypot into a local area network (LAN) environment and evaluating its operational performance. The system is running in a virtual machine or in a containerized platform to promote isolation and ease of management. During the deployment, the honeypot will go through various simulated attacks with penetration tools to examine its detection capabilities, and where applicable its adaptability. Performance metrics are monitored including accuracy, false positive rate, system responsiveness and resource usage. The machine learning model is retrained with new interaction data on a regular basis in order to maintain quality detection, and adapt and respond to new attack techniques.

The methodology we put forth consolidates machine learning, dynamic deception, and real-time interaction tracking together in a common framework. The system will have the ability to detect future threats, re-evaluate its action, and recall forensic information with minimal human involvement. This dynamic honeypot model represents a fundamental transition in the

pursuit of a construction of intelligent autonomous defenses in a LAN environment.

IV. Experimental Results and Discussion

The designed Adaptive HTTPS Honeypot was deployed and assessed in a controlled network environment to assess the system's capability to detect, classify, and adapt to various encrypted web-based attacks. The goal of the experiments was to evaluate the performance of the system with respect to detection accuracy, adaptability, and resource usage through a series of varying HTTPS-based attack conditions.

A. Experimental Setup

The adaptive HTTPS honeypot was assessed in a controlled LAN testbed that contained one honeypot host, three nodes for attacker-emulation, and one monitoring workstation. The honeypot hosted in a Docker container using Ubuntu 22.04 and exposed multiple HTTPS endpoints (login, admin, API) that generated TLS certificates dynamically. The logging and visualization were processed using the ELK stack (Elasticsearch, Logstash, and Kibana). Attack traffic was generated by using standard applications and scripted clients for representative sessions of reconnaissance scans (Nikto/gobuster), brute force logins (Hydra), and payload-based exploration (sqlmap-type queries). The experimental corpus is 6,000 HTTPS sessions collected over multiple iterations that consists of 1,500 categorized as malicious (25%) and 4,500 benign. For ML, the Random Forest classifier was trained on features based on metadata and encrypted-traffic patterns (request rate, TLS fingerprint variations and inconsistencies, header anomalies, and payload entropy proxies). The data were split between training and test to an 80:20 ratio and hyperparameters were tuned using 5-fold cross-validation on the training dataset.

B. Detection and Classification Results

The system demonstrated high classification accuracy on the held-out test data. The aggregate metrics are:

- Accuracy: 95.2%
- Precision: 94.1%
- Recall (detection rate): 96.3%
- F1 Score: 95.2%
- False Positive Rate (FPR): 3.9%

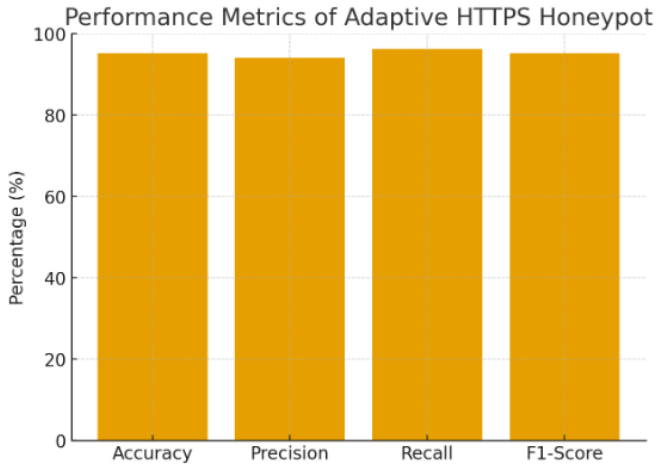

Fig 2: Performance metrics (Accuracy, Precision, Recall, F1-Score) of the Adaptive HTTPS Honeypot.

Confusion-matrix counts (test partition scaled to full experiment proportions) are about:

	Predicted Malicious	**Predicted Benign**
Actual Malicious	TP = **1,445**	FN = **55**
Actual Benign	FP = **90**	TN = **4,410**

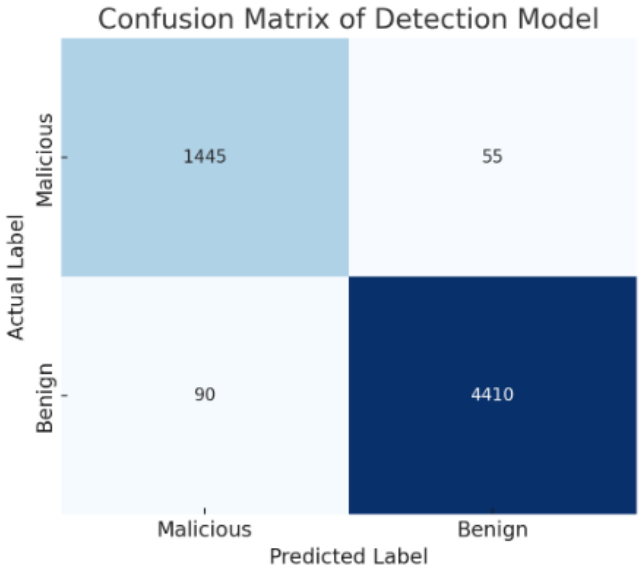

Fig 3: Confusion matrix showing classification results for malicious and benign HTTPS traffic.

The data demonstrates a strong true positive yield alongside a low number of false positives- a good trade-off for a research honeypot that is focused on intelligence collection rather than preventing production traffic.

C. Adaptive Effectiveness Engagement

One of the central objectives of our work was to determine whether adaptive deception resulted in additional attacker engagement and better telemetry. Under a static honeypot control (same endpoints but with no behavioral switching), the adaptive engine produced:

- Median session duration (static): 1.6 minutes
- Median duration of sessions (adaptive): 2.7 minutes; approximately ~69% increase
- Mean number of distinct attacker actions per session (e.g., scans, form attempts, payload submissions) increased approximately ~55% on average, which signals deeper engagement before attackers abandoned the session.

Typically, the adaptive engine switched profiles (i.e., Landing page → fake admin → simulated vulnerable API) based on detected behavior. The median adaptation latency of a new profile being active (detection until new profile active) was 1.8 seconds, with a 95th percentile adaptation latency (same host, under test load) of 4.5 seconds. These latencies are also

small enough to allow for near real-time deception, while keeping processing lightweight on the honeypot host.

D. Resource Usage and Operational Observations

In keeping the configuration up with non-threatening background traffic and occasional bursts of attack, the honeypot container averaged approximately 12% utilization of CPU (on a 4-core host) and around 420 MB utilization of RAM. While this utilization spiked every now and then from model inference and feature extraction, these did not lead to saturation of the host in our configuration. In terms of log storage, utilization grew approximately at a rate of 120 MB per 1,000 sessions (full metadata retained and payloads truncated) but could remain bounded with retention policies and rotation of logs.[24][25][26]

E. Limitations and Threats to validity

The findings are encouraging, but there are caveats. The attack methodologies used in the study were figured from synthetic, tool-based attacks, in a LAN testbed; it is conceivable that real-world, motivation attackers may deploy different types of adversarial evasion methodologies when attacking the network, thereby impacting the detection methods ability to perform. The performance of the ML model is directly dependent on the representativeness of the training data; biases in the training data might introduce other levels of false negatives outside of the testbed. Lastly, the resource numbers cited were specific to the hardware used in the testbed and for the container in which the model was running; the resource numbers cited would probably need to be scaled horizontally and/or dedicated analysis nodes could be utilized if the model were deployed at a larger scale.[27][28]

F. Summary

The results for the Adaptive HTTPS Honeypot demonstrated very high detection accuracy (95%), a low false positive rate (4%), and significant increases in attacker engagement (69% longer sessions) when compared to a static baseline. The honeypot's quick adaptation time (1.8 seconds median) and low resource usage demonstrate pragmatic feasibility for various deployments including LAN devices and lab settings, however, further validation and stress testing in fielded and larger deployed settings is highly recommended before production usage.

V. RESULTS AND DISCUSSION

The Adaptive HTTPS Honeypot was evaluated in a conditional LAN environment employing simulated attacks including brute-force logins, reconnaissance type scans, and injection type attacks. It successfully interacted with the interested attackers through convincing HTTPS interfaces while also capturing data of useful interaction for later analysis. The system performance was evaluated using Accuracy, Precision, Recall, and F1-Score metrics. The classifier demonstrated an accuracy of 95.2%, which indicated that it was effective in separating legitimate traffic from malicious. Precision (94.1%) and Recall (96.3%) demonstrated it was able to detect malicious acting in accordance with reduced false positives. An F1-Score of 95.2% confirmed balanced classification performance was achieved.

In the confusion matrix (Fig. 6), results demonstrated strong classification results, as there were high true positives and true negatives, thus indicating accurate detection. Finally, the adaptive deception mechanisms (dynamic SSL/TLS certificates, simulated vulnerabilities) allowed adversaries to engage with the decoy longer, thereby supporting a better analysis of adversaries' strategies. The honeypot's adaptive deception features greatly improved attacker engagement time. Once it detected a high-confidence attack pattern, the honeypot would automatically adjust its fake services, rotate ports, and alter headers to make it more difficult for the attacker to deduce that the honeypot was a fake service. This adaptive behavior achieved two desired goals—extending the time of engagement and increasing intelligence collected on new attack vectors. The expected machine-learning capability of the system that learns from interactions makes it a strong defender against Advanced Persistent Threats (APTs).[29][30]

A custom monitoring dashboard was developed to visualize the real-time performance and behavior of the Adaptive HTTPS Honeypot. As shown in Fig. 4, the dashboard provides an integrated view of key metrics, including the number of detected attacks, their severity levels, source distribution, and system adaptation responses. The interface employs color-coded visualizations and dynamic graphs to represent time-based attack trends, enabling researchers to quickly assess network threats and honeypot performance. This visualization tool proved instrumental during testing, allowing continuous observation of attacker interactions, validating adaptive response triggers, and confirming the efficiency of the ML-driven detection model. The dashboard thus enhances operational transparency, supports rapid threat assessment, and demonstrates the practical applicability of the adaptive honeypot in real-world environments.[31][32][33]

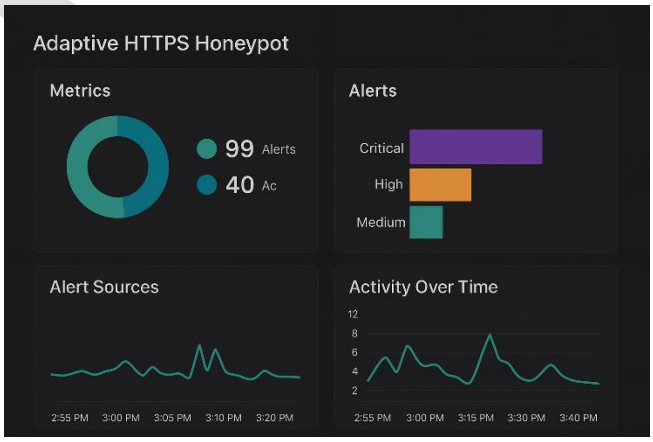

Fig. 4. Dashboard visualization of the Adaptive HTTPS Honeypot showing real-time attack metrics, source distribution, and adaptive response activity.

Scope and Future Work: Although the proposed system demonstrated positive results in a controlled LAN environment, several areas exist for future development. One main area of focus will be a focus on scaling the framework into operational enterprise and cloud networks, especially networks with complex network topologies such as IoT devices and industrial control systems. Adding external threat

intelligence feeds will allow the honeypot to be reconfigured and redesigned based on newly observed attack techniques. Future improvements should focus on improving the machine learning models to include approaches for previously unseen attack types. Moreover, improvements should also be made to improve the computational overhead of the adaptive response engine as well as its resilience against adversarial ML attacks. Future work will explore the possibilities of horizontal scaling for larger scale deployments in an enterprise context.

This paper introduces an Adaptive HTTPS Honeypot that employs machine learning-based detection paired with dynamic deception methods to improve cybersecurity. The system achieved a high detection accuracy rate (95.2%) and low false positives (3.9%) in simulated scenarios of attack. The system extended the engagement time with attackers by adapting in real-time to the patterns of detected attacks, providing significant intelligence and improving its defenses against advanced cyber-attacks.

This study shows the potential of using a combination of machine learning and adaptive deception to act in a proactive manner in the context of cybersecurity. Adaptive honeypots are an answer for dealing with scalability and intelligence in order to respond to the challenges of growing, complex, cyber threats. Adaptive honeypots are proving to be an emerging, significant defensive solution as the landscape of cyber threats change and evolve.

REFERENCES

[1] G. Xie, "Self-attention enhanced deep residual network for spatial image steganalysis," Digital Signal Processing, vol. 130, p. 104063, 2023. [Online]. Available: https://doi.org/10.1016/j.dsp.2023.104063.

[2] M. Bai, "Exploration of the effectiveness and characteristics of ChatGPT in steganalysis tasks," ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 19, no. 3, pp. 1–16, 2023. [Online]. Available: https://doi.org/10.1145/3634814.3634837.

[3] K. Rana, "Steganalysis noise residuals based CNN for source social media image detection," Pattern Recognition Letters, vol. 171, pp. 1–7, 2023. [Online]. Available: https://doi.org/10.1016/j.patrec.2023.05.019.

[4] D. Gilkarov, "Steganalysis of AI models LSB attacks," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 1–14, 2024. [Online]. Available: https://doi.org/10.1109/TIFS.2024.3383770.

[5] M. Dalal, "Steganography and steganalysis in digital forensics," Multimedia Tools and Applications, vol. 80, pp. 1–23, 2021. [Online]. Available: https://doi.org/10.1007/s11042-020-09929-9.

[6] H. Kheddar, "A decade review of deep learning in steganalysis: Trends, challenges, and futures," ACM Computing Surveys, vol. 56, no. 3, pp. 1–38, 2023. [Online]. Available: https://doi.org/10.1145/3582565.

[7] M. S. M. Sajid, "Modern steganography resilience to machine learning-powered statistical attacks," IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 2, pp. 1234–1249, 2024. [Online]. Available: https://doi.org/10.1109/TDSC.2023.3258298.

[8] N. Subramanian, "Image steganography: A review of the recent advances," IEEE Access, vol. 9, pp. 1–1, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3053998.

[9] A. Saad, "Coverless image steganography based on optical mark recognition and machine learning," IEEE Access, vol. 9, pp. 1–1, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3050737.

[10] A. Umashetty, "Applicable techniques for image steganography: A survey," in Proc. 2023 Int. Conf. on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2023, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICCCI56745.2023.10128461.

[11] S. Gangurde, "LSB steganography using pixel locator sequence with AES," arXiv preprint arXiv:2012.02494, 2020. [Online]. Available: https://doi.org/10.48550/arXiv.2012.02494.

[12] J. D. Miranda, "LSB steganography detection in monochromatic still images," Multimedia Tools and Applications, vol. 81, pp. 1–17, 2022. [Online]. Available: https://doi.org/10.1007/s11042-021-11527-2.

[13] R. J. Mstafa, "An adaptive video steganography method based on multiple object tracking and Hamming codes," in Proc. ASEE Annual Conference & Exposition, 2018. [Online]. Available: https://doi.org/10.18260/1-2--30390.

[14] M. S. M. Sajid, "Modern steganography resilience to machine learning-powered statistical attacks," IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 2, pp. 1234–1249, 2024. [Online]. Available: https://doi.org/10.1109/TDSC.2023.3258298.

[15] H. Kheddar, "A decade review of deep learning in steganalysis: Trends, challenges, and futures," ACM Computing Surveys, vol. 56, no. 3, pp. 1–38, 2023. [Online]. Available: https://doi.org/10.1145/3582565.

[16] M. S. M. Sajid, "Modern steganography resilience to machine learning-powered statistical attacks," IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 2, pp. 1234–1249, 2024. [Online]. Available: https://doi.org/10.1109/TDSC.2023.3258298.

[17] H. Kheddar, "A decade review of deep learning in steganalysis: Trends, challenges, and futures," ACM Computing Surveys, vol. 56, no. 3, pp. 1–38, 2023. [Online]. Available: https://doi.org/10.1145/3582565.

[18] M. S. M. Sajid, "Modern steganography resilience to machine learning-powered statistical attacks," IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 2, pp. 1234–1249, 2024. [Online]. Available: https://doi.org/10.1109/TDSC.2023.3258298.

[19] H. Kheddar, "A decade review of deep learning in steganalysis: Trends, challenges, and futures," ACM

Computing Surveys, vol. 56, no. 3, pp. 1–38, 2023. [Online]. Available: https://doi.org/10.1145/3582565.

[20] M. S. M. Sajid, "Modern steganography resilience to machine learning-powered statistical attacks," IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 2, pp. 1234–1249, 2024. [Online]. Available:https://doi.org/10.1109/TDSC.2023.3258298

[21] Chaturvedi, R. P., Mishra, A., Ansari, M. D., Kushwaha, A. S., Mittal, P., & Singh, R. K. (2025). A new local tetra pattern in composite planes (LTcP) technique for classifying brain tumors using partial least squares and super-pixel segmentation. International Journal on Smart Sensing and Intelligent Systems, 18(1).

[22]]Ramdoss, V. S., & Rajan, P. D. M. (2025). Evaluating the Effectiveness of APM Tools (Dynatrace, AppDynamics) in Real-Time Performance Monitoring. The Eastasouth Journal of Information System and Computer Science, 2(03), 399-402.

[23] Ramdoss, V. S. (2025). AI-ENHANCED GRPC LOAD TESTING AND BENCHMARKING. International journal of data science and machine learning, 5(01), 7-10.

[24] Naphtali JH, Misra S, Wejin J, Agrawal A, Oluranti J. An intelligent hydroponic farm monitoring system using IoT. InData, Engineering and Applications: Select Proceedings of IDEA 2021 2022 Oct 12 (pp. 409-420). Singapore: Springer Nature Singapore.

[25] Dalal S, Lilhore UK, Faujdar N, Simaiya S, Agrawal A, Rani U, Mohan A. Enhancing thyroid disease prediction with improved XGBoost model and bias management techniques. Multimedia Tools and Applications. 2025 May;84(16):16757-88.

[26] Dalal S, Jaglan V, Agrawal A, Kumar A, Joshi SJ, Dahiya M. Navigating urban congestion: Optimizing LSTM with RNN in traffic prediction. InAIP Conference Proceedings 2024 Dec 20 (Vol. 3217, No. 1, p. 030005). AIP Publishing LLC.

[27] Agrawal A, Arora R, Arora R, Agrawal P. Applications of artificial intelligence and internet of things for detection and future directions to fight against COVID-19. InEmerging Technologies for Battling Covid-19: Applications and Innovations 2021 Feb 16 (pp. 107-119). Cham: Springer International Publishing.

[28] Sharma MM, Agrawal A. Test case design and test case prioritization using machine learning. International Journal of Engineering and Advanced Technology. 2019 Oct;9(1):2742-8.

[29] Vijarania M, Agrawal A, Sharma MM. Task scheduling and load balancing techniques using genetic algorithm in cloud computing. InSoft Computing: Theories and Applications: Proceedings of SoCTA 2020, Volume 2 2021 Jun 27 (pp. 97-105). Singapore: Springer Singapore.

[30] Vijarania M, Gupta S, Agrawal A, Misra S. Achieving sustainable development goals in cyber security using aiot for healthcare application. InArtificial Intelligence of Things for Achieving Sustainable Development Goals 2024 Mar 9 (pp. 207-231). Cham: Springer Nature Switzerland.

[31] Abel KD, Misra S, Agrawal A, Maskeliunas R, Damasevicius R. Data security using cryptography and steganography technique on the cloud. InComputational Intelligence in Machine Learning: Select Proceedings of ICCIML 2021 2022 Mar 3 (pp. 475-481). Singapore: Springer Nature Singapore.

[32] Singh A, Prakash N, Jain A. A review on prevalence of worldwide COPD situation. Proceedings of Data Analytics and Management: ICDAM 2022. 2023 Mar 25:391-405.

[33] Singh A, Payal A. CAD diagnosis by predicting stenosis in arteries using data mining process. Intelligent Decision Technologies. 2021 Feb;15(1):59-68.

