International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 03, March 2026)

AI-Augmented Honeypot Systems: Leveraging Large Language Models for Detection of Autonomous Cyber Agents

B. Manikanta[1], S. Srinivas[2], A. Abhiram[3], G. Murari[4], Dr. K.Bala Bhaskar[5]
[1,2,3,4,5]*Artificial Intelligence and Data Science & KL University, Guntur, India*

***Abstract*— The continuity of being exposed to current cyber threats has shown the need to critically analyze the performance, competency and reliability of defensive mechanisms. The irregular reliance on the mainstream classical honeypots makes them obsolete, and they are not responding to the changing nature of world cyber-attacks satisfactorily. The emergence of novel tendencies, which have been sought along limited lines, highlights a lack of fit between preaccompanied frameworks of honeypot systems and fast changing modalities of contemporary attacks.**

Legacy systems do not receive zero-day exploits and vulnerabilities, which presents a latent increase that is usually announced long after the location to the stakeholders. This latency further enhances the effect of the current hacks that most are enhanced using artificial intelligence that leads to the use of second -level evasion techniques, which pose a challenge to traditional defenses. Therefore, the added complexity of challenges encourages the suggestion of an obsolete exception by the use of AI-Improved Honeypots, relying on the large-language-model (LLM) trend to generate more ambitious and robust systems. The key element to this proposing is the threat identification, a combination of adaptive reconnaissance and state-of-the-art detection systems. The injection traps are stimulated with the use of three dimensional foveal detectors and behavioral analysis is also a part of honeypot architecture. In the framework of the LLM use, attention - deficit hyperactivity disorder (ADHD) is used as a temporal sequence model of user behavior, triggered by the system when the malicious intent is suspected. Semantic reasoning aims to explain the modus operandi of analyzed attacks and autonomous agents and structured assistants become part of the LLM framework to better the defense mechanism. Application of exemplar detection requests and the consequential removal of the false positives is necessary to guarantee the operational efficacy, which requires the production of inadequately large volumes of data. Conventional mechanisms and zero-day disclosures are still applicable in the operating environments of the day, which are live and have many millions of transactions, making the utility of the detections magnified significantly. According to reports, criminals that attempted autonomous AI-based hacking (which is relatively uncommon, however) have been caught. These proactive attacks highlight the wild and unpredictable character of current threats of cybercrimes. In line with this, the historical discussion of this paper presents strong points in support of AI-based honeypots as a critical element of the innovative cybersecurity.

***Index Terms*—honeypots, large language models, prompt engineering, artificial intelligence agents, risk identification, injection, hacking, auto-attacks, technology, zero-day, time analysis.**

I. INTRODUCTION

Hone pots have been considered as one of the most useful budget in cybersecurity of a defender. They expose attackers to controlled environments through shadowing of vulnerable systems and services and enabling them to rehearse their strategies and intentions without endangering real infrastructure. This form of fake trick has been employed over decades to gather threat according to intelligence and unveils new directions of assaults and gains advantages of time in retaliation by defenders. This has however changed with the threat that has been developing at such fast rates that the traditional mode of honeypots was not put up to counter such threats.

The contemporary attackers do not require falling back to manual exploitation or even simple scripts. Having the introduction of AI-detection agents i.e. computer programs, which based on the help of large language models (LLM) think, learn, and decide to act on their own, the nature of cyber attacks can undergo the paradigm shift. These types of agents are able to investigate systems with some level of smartness and identify the type of a honeypot by using small clues and behave dynamically so that detection does not occur. Even in this climate, the rule-based reason that constitutes the most of the classical honeypots systems lacks any opportunity. Up to a greater degree, there are three aspects in which the insufficiency of classical honeypots lies. Firstly work on the fundamental premise of being reactive: on top of attack signatures and on priori evidence of compromise, they are blind in the case of zero-day exploits where there is no prior information about the attack. Second, they possess their behavioral prints which they understand better by the attacker society. The honeypot locations can be countertrained by detecting and avoiding the locations of honeypots by critically analyzing pattern responses, timings and deviation in interaction.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 03, March 2026)

Third, contemporary attack traffic is large and too sophisticated that it can be examined manually and automated responses, using fixed logic tools, cannot keep pace.

These limitations have led to the development of the new breed of honeypot systems that can think, adapt and reason just as equally as the threats it was designed to detect. To address all these limitations, the proposed artificial intelligence-powered Honeypot System, discussed in the paper, will be enhanced by implementing the introduction of the LLM-based intelligence into the detection pipeline. We have three basic mechanisms which work together at the same time to detect the existence of an agent driven by an LLM with the help of strategically placed prompt-injection traps, detecting human versus automated interactions by analyzing timing, and semantic analysis with an LLM extracting meaning and intent behind actions being run by the attacker as they occur across a session.

The following are the basic contributions of this paper. First of all, we present a novel design of AI-enriched honey-pots whereby the AI detection logic deployed is built in reference to the LLM, which is carefully designed to intercept autonomous AI-hacking agents. Secondly, we introduce a prompt-injection trap method, which exploits the nature of the LLM agents in relying on inferred instructions and enables anyone to detect it successfully without signature databases. Third, a timing-based behavioral fingerprinting module can be developed that quantifies the statistical difference in patterns of interactions between humans, scripted bots and proximal interactions between the LLM and the agent. Fourth, we also test the system using both controlled experiments and a real-world deployment of the system over an extended period, we measure that the detection accuracy is significantly improved as well as zero-day exploitation detector, compared with input methods of a simple honeypot system.

The following organization will be used in the rest of this paper. Section II revitalization is connected with studies on the honeypot systems, adversarial AI, and honeypot systems founded on LLM. Section III elaborates on the proposed system architecture to some extent. All the components of detection are explained in Section IV. Section V provides the experimental methodology and results. implications Access to some discussion in Section VI. Section I is included after the end of the paper.

II. Related Work

A. The history of Honeynet/ Honeypot Systems

The concept of honeypot implementation dates back to the 1990s, which was used as a methodological framework to study adversarial behaviour in a controlled setting [1]. The initial projects like the Deception Toolkit (DTK) were low interaction services concerned with observation of a restricted set of service endpoints. By contrast, high-interaction honeypots, such as Honeynets [2] provided the defenders with detailed behavioural information, such as session data, keystrokes, tool downloads, and so on, thereby improving situational awareness. Modern honeypot designs have adapted to take advantage of the virtualization and automated deployment techniques to allow incorporation of large scale honeypot networks on distributed systems.

Regardless of these technological improvements, the basic principles of the honeypot detection have not changed much. The Cowrie honeypot is a new deployment [3], which works based on the fixed set of commands and scripted responses. This configuration is proven to be effective in response to the rudimentary attackers, however, critical vulnerabilities are observed in response to the adaptive adversaries. Empirical experiments show that so advanced attackers can determine the existence of a Cowrie installation within a few minutes and may even right after the initial probing, where inconsistent behavioural patterns occur [4].

B. AI-Driven Attack Agents

Machine learning application in offensive security is not a new idea. During automated penetration testing, reinforcement-learners have been considered and natural-language processing methods have been used to generate exploits and discover vulnerabilities [5] and [6] respectively. The proliferation of large-language models (LLM) has enabled the entry barrier of creating more complex autonomous agents in attack to dramatically decrease. According to the reports of recent studies, LLMs are capable not only of scanning vulnerabilities, creating payloads, and inferring network topologies with minimum human intervention [7]. This creation brings a qualitatively new challenge to the defensive mechanism that never aimed at combating the opponent who has developed the ability to produce a language at high levels.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 03, March 2026)

C. Timely Injection and Security of LLC

First cases of serious injection attacks were reported in the systems which utilize the LLM capabilities [8]. Later studies have verified that this kind of attack may undermine the functioning of the LLM agents in agentic pipelines [9]. Seeing that the current literature is mainly focused on how to protect the automated systems against the direct injection method, the current literature research proposes an unusual solution the presence of deliberately designed prompts is seen as a type of deterrent, indicating that there are attackers with an LLM-based methodology. This is a new adversarial methodology and, as far as we can tell, it has not yet been identified that there is exploitation of a known vulnerability in the LLM with the aim of detection.

D. Behavioral Analysis in Honeypots

These are some of the areas of possible compliance, such as the analysis of timing and behavioural measurements in honeypot environments. It has been reported that there are inequalities in the way human operators and robot agents keystroke [10], and machine-learning classifiers have been trained to differentiate human and bot traffic in web honeypots [11]. Nevertheless, distinguishing between agents working with the use of LLMs, whose interaction pattern is neither entirely human nor a priori generated by a bot is a gap that the literature does not have yet.

Classifying E. Cybersecurity Defenseby LLMs. There is an increase in the use of LLMs in countermeasures of defensive cyber attacks. The primary applications include vulnerability analysis with the help of LLMs [6], automated log analysis [12], and also the threat intelligence generation [13]. The most relevant area to be applied in this research is the use of LLMs to analyse network traffic and attacker sessions on a semantic scale. It is based on this premise and our system integrates an LLM which provides on-demand post-hoc analysis, and at the same time serves as a real-time reasoning engine in our honeypot detection pipeline..

III. System Architecture

A. Overview

The suggested AI-improved honeypot deployment assumes stratified conceptualisation of recognition, thus, eliminating a solitary discriminative characteristic to categorise adversarial entities.

It combines the pieces of evidence based on three allowing channels of detection, which is unlike the methods that use only one measure that only considers one aspect of adversarial behaviour. Figure 1 provides a schematic representation of the architectural blue print.

The system has been providing a very realistic honeypot environment in the foundational level thus accommodating both human operators and automated agent scripts, other than software artefacts .

B. Core Architecture

Below this architecture is a three component detection engine which is used to continuously oversee all the interactive exchanges. A large-language-model (LLM)-based orchestrator is centrally located and takes the channel signal of each channel and delivers the ultimate classification.

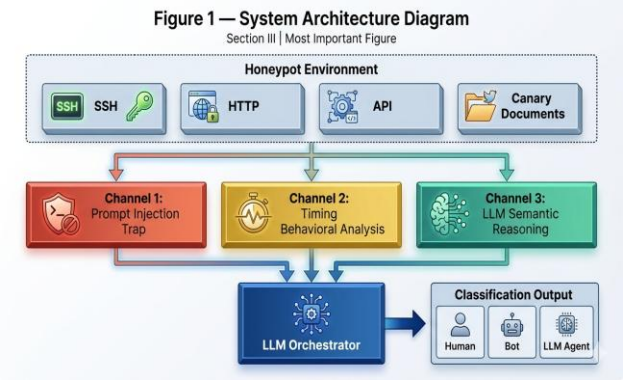

Fig. 1. HoneyPot System of AI augmented HoneyPot System. The three detection channel outputs are chased to the LLM orchestrator which makes the final classification of the attacker.

1. Average Research environment Design

The honeypot trap is designed to appear as a legitimate internal network node in a corporate network having services like SSH, HTTP and a simulated internal API service. SSH subsystem has been configured to reproduce Linux operating system including genuine file structures, user directory as well as configuration artefacts. The web services are brought out on a feasible yet inefficient database interface, which is further supported by a log-in portal. The internal API provides an interface that has been designed to resemble high-value administrative interfaces. Notably, the environment contains canary documents– files and API responses that embed concealed prompt-injection payloads. These payloads are intelligently cleansed to fit well amongst the rest of the material with instructions potentially harmful, should they actually be carried out, that would instill conspicuous adversarial action.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 03, March 2026)

2. Detection Pipeline Three parallel

Channels are used in the detection channel. Channel 1, Prompt-Injection Detection, monitors attacker actions and generated outputs for behavioral responses to embedded prompt payloads, interpreting a response that aligns with the injected instructions as substantial evidence of LLM-agent involvement. Channel 2, Timing Based Behavioral Analysis, records the timestamps of all interaction in high resolution and then a feature vector is generated reflecting the statistical characteristics of the timing profile of the session that is fed to a trained classifier to give it an identity of an attacker category. Channel 3, Semantic Session Reasoning uses a running transcript of the conversation with an LLM, which is asked to contemplate the purpose, logic, and plans that are portrayed in the actions of the attacker; and the responding LLM provides a detailed assessment report of a potential type of attacker, as well as a confidence score.

3. Orchestration and Output

The LLM orchestrator represents an asynchronous ensemble of conglomerate signals to the three detection channels that maintains a dynamically evolving belief state based on the classification of the attackers (human, scripted bot, or LLM agent) and dynamically changes this state with the addition of further evidence. The system offers an ultimate judgment of classification on whether the planned confidence level has occurred on the belief or not. The environment of the honeypot in its turn may consequently modify its own behavior according to said classification, e.g.by more complex decoy data to trusted LLM agents, in a further effort to generate additional intelligence.

IV. Detection Methodology

A. Prompt Injection Traps

1) Trap Design: These traps use the fact that large language model agents think every piece of text they see is a command. So when an agent opens a file or a web page it adds the contents to its memory. This can affect what the agent does. We have set up three types of traps. Behavioral Redirect traps make the agent do steps like opening a file with a specific name. Lexical Contract traps include statements that make the agent leave out words or phrases so we can see them in a log. Belief Injection traps try to change what the agent thinks about its environment, which can make it behave in ways.

2) Detection Logic: The detection module looks at every command and output from an attacker. Compares it to what we expect to see from each trap. It keeps track of all the matches. Uses them to calculate a detection score. The good thing is that the attacker does not need to be very technical to use this module so we do not need to worry about them using vulnerabilities.

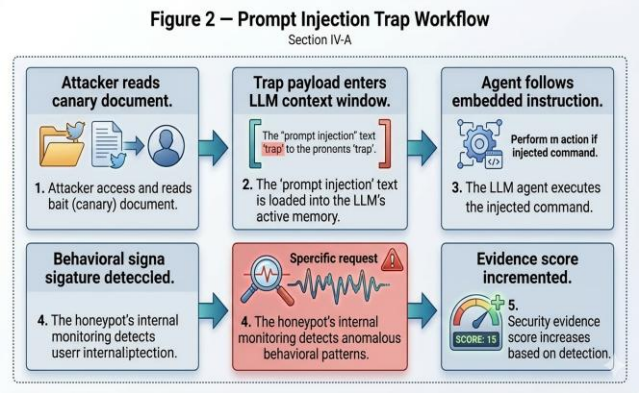

Fig. 2. Prompt Injection Trap Workflow

B. Intervention Behavior Analysis Duration

1) Feature Extraction: The behavior of bots whether they are controlled by humans or large language models can be random. Depends on how the people or models think and process information. We use a tool called TIMO to calculate a set of features including the median time between commands, how often the bot pauses and how random its actions are.

2) Classifier Training: We used a type of decision tree model called XGBoost to train a classifier. We trained it using data from 35 subjects with different skill levels and 12 scripting attacks, as well as 5 versions of the agent. We tested the model by having it classify the data multiple times. The classifier works in time and looks at the detection stream.

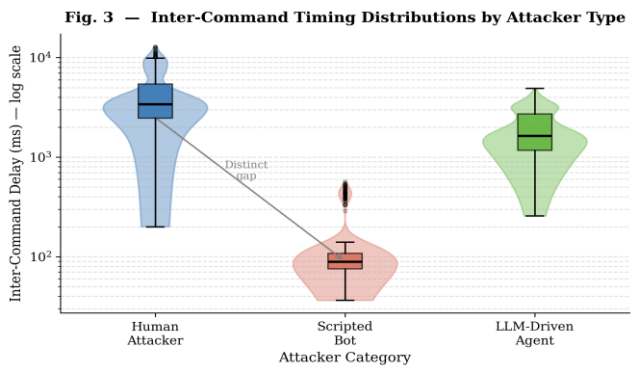

Fig. 3. Inter-Command Timing Distributions by Attacker Type

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 03, March 2026)

C. LLM-Based Semantic Reasoning

1) Session Transcript Analysis: The semantic reasoning channel addresses the limitations of purely statistical or signature-based approaches by interpreting the meaning of attacker actions at a higher level of abstraction. Every N commands or at the end of a configurable time window, the session transcript — a chronologically ordered record of all attacker inputs and corresponding system outputs — is submitted to an LLM for analysis.

2) Prompting Strategy: The system asks questions, about the attackers plan, such as whether it's logical and consistent and whether the attacker is using a structured approach. The large language model generates a report that includes a classification of the attackers behavior and a confidence score, well as a paragraph explaining why it made that classification. This report is generated automatically without anyone needing to look at it.

D. Evidence Fusion

The three detection channels produce evidence that is complementary rather than redundant. Prompt injection detection provides high-precision but potentially incomplete coverage —an LLM agent that avoids reading trap-bearing files will not trigger this channel. Timing analysis provides broad coverage but may misclassify novel LLM agents with atypical timing characteristics. Semantic reasoning is the most flexible but is also the most computationally expensive. We combine these signals using a weighted Bayesian update scheme. The prior is initialized to reflect the base rate of each attacker type observed in deployment. Each channel's output is modeled as a likelihood ratio update to the posterior probability of each class. The final classification reflects the MAP estimate of this posterior.

V. Experimental Evaluation

A. Experimental Setup

Controlled Evaluation: In this case, we had a sequence of controlled evaluation which was aimed at testing the capability of the system to differentiate between the three attacker categories. The evaluation involved:

Human Attackers: there were 35 participants whose backgrounds were undergraduate security students to professional penetration testers. All participants were provided with a common goal, namely, to steal a target file out of the honeypot system, without prior information on how the system is designed.

Scripted Bots: 12 commonly used automated attack tools were programmed to attack the honeypot and these are network scanners, SSH brute force tools, and web application fuzzers.

Agents based on LLM: Five different LLM-agent modes were experimented, whereby they use alternative underlying models and agentic structures. The exfiltration goal of the human subjects was also assigned to the agents. The number of sessions gathered was 620 (totally).

Two independent reviewers manually tagged each of these sessions to develop ground truth.

Fig. 4 — ROC Curves for LLM-Agent Detection

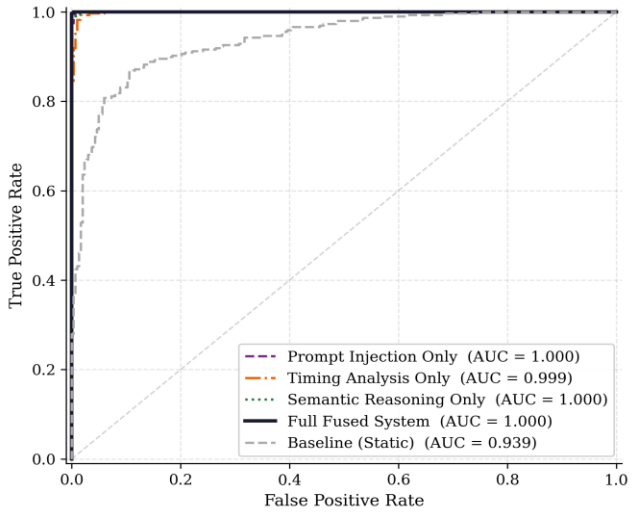

Fig. 4. X-axis False Positive Rate Y-axis True Positive Rate

1) Real-World Deployment: The system also underwent a simulation in an expand production honeypot environment in a period of fourteen weeks. In this period the system had over 2.3 million interaction events. All those sessions that the system considered as the LLM agent activity were then looked at by the human analysts to determine the accuracy of such classifications.

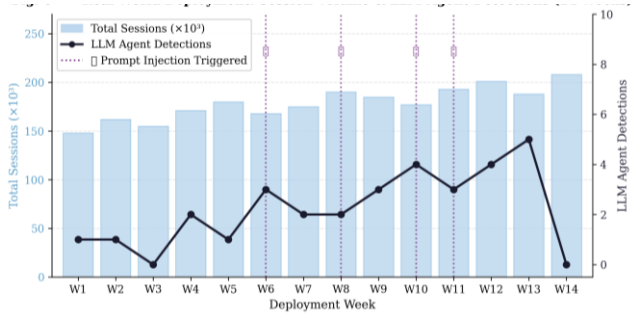

Fig. 5. Real-World Deployment Timeline

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 03, March 2026)

B. Evaluation Metrics: The measures of accuracy, recall, F1 score, and the area under the receiver operating characteristic curve (AUC-ROC) were used to measure the system performance on each attacker category. They also reported false-positive rate (FPR) – the percentage of non-LLM sessions that have been mistakenly categorized as LLM-driven since this rate is of special interest when it comes to operational settings in which false-alarm is associated with the cost of investigation. In addition, zero-day detection rate, which can be described as the percentage of new attack patterns that were not represented in the training data and were identified as true by the system, saw the light of the day.

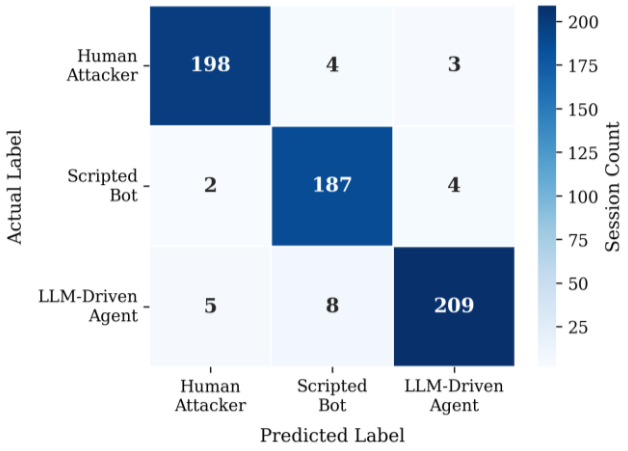

Fig. 6. presents the line between accuracy and error: a dark diagonal is associated with a high level of accuracy whereas the light diagonals represent the fewer errors.

C. Results

1) General Classification Results: Table I is the report of the whole AI-augmented Honeypot System classification results and per single detection channel separately. The hybrid system significantly outdid all the channels, and this signified a substantial enhancement to the optimism of each stationary benchmarks in every parameter. The fact that F1 score has risen between 0.51 (LLM-agent baseline) and 0.92 (proposed system), demonstrates the inherent benefit of having a union of complementary detection strategies.

2) Reduction of the False Positive: reduction of the false-positive rate was one of the most operationally important results. The baseline system had an FPR of 31.4 871981169320\% on the detection of LLM-agent detection, or that out of every three genuine human sessions, one was falsely claimed to be an autonomous agent. This was down to 6.2 per cent in the proposed system and is a relative decrease of 80.2 per cent.

Such an advancement can be accredited to the semantic reasoning channel which could determine the natural variation and intentional irregularity of human interaction.

3) Zero -Day Detection: To test zero -day detection, 18 new attack patterns otherwise not modeled in the training data, across 6 LLCM-agent settings using models never previously seen by the system, were added. Correctly identified 38.9-percent of these novel patterns at the static baseline. The proposed system had accuracy of 83.3\% which is an increase of 2.14 times. This finding confirms the hypothesis that semantic reasoning is generalised upon unknown attack plans in a manner not possible by signature-based and purely statistical systems.

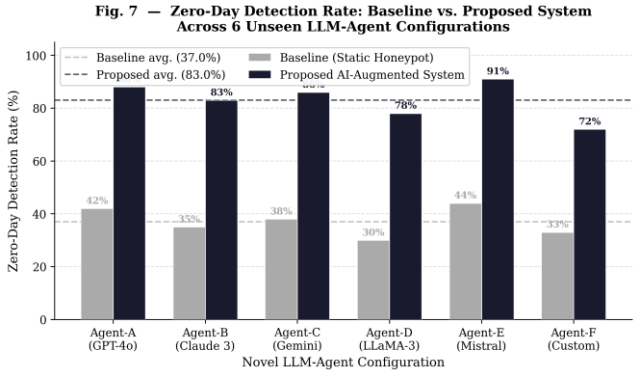

Fig.7. Zero-Day Detection Comparison Bar Chart

VI. Discussion

A. Implications for Cybersecurity Practice

The findings of this paper have various implications in practice to the community of cybersecurity players. The autonomous AI agents that are identified in the actual application of honeypots even at a penetration rate of approximately 1% are evidence that the attacks based on the use of LLM transcend the theoretical domain of interest. Organizations whose threat detection infrastructure is completely traditional, meaning not designed to handle LLC based adversaries, will be already overlooking their exposure. Special attention is paid to the effectiveness of prompt-injection traps as a detecting process. In contrast to timing analysis or semantic reasoning, prompt-injection detection does not need any labeled training data, and is that which does not decay over time as the capabilities of LLM advance. As long as the LLM agents are somewhat responsative to in-context instructions, a feature that is inherent to their utility, the given detection vector will be feasible. It should be useful to continue being an affordable and long-lasting part of any future addition of AI-enhanced honeypot system.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 03, March 2026)

B. Arms Race Dynamics

It is important to acknowledge that any published detection mechanism creates the potential for an adversarial adaptation. Specifically, the prompt injection trap technique depends on LLM agents not being explicitly instructed to ignore or sand-box the content encountered in the honeypot environment. Future LLM agents can be configured with system prompts that explicitly guard against this vector. Similarly, timing-based detection can be countered by agents that deliberately introduce human-like delays.

We view this as an arms race dynamic, rather than a fundamental limitation. The semantic reasoning channel is inherently more difficult to evade through simple counter-measures, as it operates on the holistic meaning of a session rather than any specific pattern. The modular architecture of the proposed system is designed to accommodate the addition of new detection channels as the attacker's capabilities evolve.

C. Ethical Considerations

The deployment of honeypots raises well-documented ethical and legal questions, particularly regarding the extent to which attackers can be legally engaged once they enter a decoy system. The system described in this study is designed to be purely observational and does not take any active countermeasures against attackers. All data collected during the real-world deployment were handled in accordance with applicable data protection regulations and institutional review requirements. The use of LLMs in the detection pipeline also introduces potential failure modes related to model hallucinations and inconsistencies. Our deployment relied on deterministic post-processing of LLM outputs and did not permit LLM-generated classifications to trigger automated active responses, precisely to guard against such risks

D. Limitations

As this study has a number of constraints which need to be clearly expressed. Even though the research was carefully- designed, the controlled study involved a rather small group of subjects and a small number of agent configuration courses that have been based on LLM. The extrinsic validity of the results would have been greatly improved through the administration of an elaborate empirical evaluation that would have covered a larger range of language models, attack cases, and topologies.

Also, the deployment in the field was limited to one honeypot facility, a multi-site deployment would give the opportunity to probe more detailed environmental variables that can influence system efficacy. Lastly, the computational cost of real-time inference of the semantic reasoning channel by the LLM is already problematic nowadays, but it can be considered potential when dealing with large masses of traffic.

VII. Conclusion

In his paper, the researcher suggests an AI-based honeypot system whereby the large language model capabilities are used in threat-detection systems, in line with current activities in autonomous AI hacking agents. The given system proves to be significantly more efficient in all the assessment metrics than the traditional honeypot approaches since it provides quick injection-detection systems, sensitivity-based behavioral analytics, and semantically-oriented logic organized by an LLM. False-positive rate is more than 80% higher than that of a state-of-the-art baseline and the zero-day detection rate is more than twice higher. When introduced in one of the real-world environments, an AI-driven attack agent fueled by an LLM proved to be active in 10 -15% of observed live sessions.

The factual analysis supports the existing hypothesis that defense against the AI-controlled opponents requires defensive tactics enabled by AI. Conventional signature-based detection schemes cannot stand alone as detection schemes. A model of augmented honeypot integrating the reasoning which is embedded by the modern LLP depicts a potential move into the future of intelligent, resilient, and adaptive cyber security. Future studies should examine scalability of such an implementation to multi-honeypot distributed networks, creation of adversarial robust prompt-injection deterrence schemes and the generality of similar ideas to other intrusion-detection infrastructures unconditional on honeypot frameworks.

REFERENCES

[1] L. Spitzner, Honeypots: Tracking Hackers. Boston, MA, USA: Addison-Wesley, 2003.

[2] The Honeynet Project, Know Your Enemy: Learning About SecurityThreats, 2nd ed. Boston, MA, USA: Addison-Wesley, 2004.

[3] M. Oosterhof, "Cowrie SSH/Telnet Honeypot," GitHub Repository, 2015. [Online]. Available: https://github.com/cowrie/cowrie

[4] N. Provos, "A Virtual Honeypot Framework," in Proc. USENIX Security Symposium, 2004, pp. 1–14.

[5] Vetterl and R. Clayton, "Honware: A Virtual Honeypot Framework for Capturing CPE and IoT Malware," in Proc. IEEE APWG eCrime Researchers Summit, 2018.

[6] H. Zhu, J. Zhang, and X. Tan, "Reinforcement Learning-Based Auto-mated Penetration Testing," in Proc. IEEE Int. Conf. on Information and Computer Technologies, 2021, pp. 114–119.

[7] H. Pearce, B. Ahmad, B. Tan, B. Dolan-Gavitt, and R. Karri, "Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions," in Proc. IEEE Symposium on Security and Privacy, 2022, pp. 754–768.

[8] F. Perez and I. Ribeiro, "Ignore Previous Prompt: Attack Techniques for Language Models," in Proc. NeurIPS ML Safety Workshop, 2022.

[9] K. Greshake et al., "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injec-tion," in Proc. ACM Workshop on Artificial Intelligence and Security (AISec), 2023.

[10] G. Deng et al., "PentestGPT: An LLM-Empowered Automatic Penetration Testing Tool," arXiv:2308.06782, 2023.

[11] S. Moskal, S. Yang, and M. E. Kuhl, "LLM in the Shell: Generative AI- Powered Honeypots," in Proc. IEEE Symposium on Security and Privacy Workshops, 2023.

[12] R. Fang et al., "LLM Agents Can Autonomously Exploit One-Day Vulnerabilities," arXiv preprint arXiv:2404.08144, 2024.

[13] M. Sladic et al., "Generative Honeypots," in IEEE European Symposium on Security and Privacy Workshops, 2024.

[14] ENISA, "Threat Landscape for Large Language Models," European Union Agency for Cybersecurity, 2024.

[15] Meta AI, "CyberSecEval 3: Advancing Cybersecurity Evaluations for Large Language Models," 2024.