IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 July 2025, accepted 27 July 2025, date of publication 4 August 2025, date of current version 8 August 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3595424

RESEARCH ARTICLE

From Prompts to Motors: Man-in-the-Middle Attacks on LLM-Enabled Vacuum Robots

ASIF SHAIKH, AYGÜN VAROL, AND JOHANNA VIRKKI
Faculty of Information Technology and Communication Sciences, Tampere University, 33100 Tampere, Finland
Corresponding author: Asif Shaikh (asif.shaikh@tuni.fi)

This work was supported by the Jane and Aatos Erkko Foundation (EVIL-AI Project).

ABSTRACT The integration of large language models (LLMs) into robotic platforms is transforming human–robot interaction by enabling more natural communication and adaptive task execution. However, this advancement also introduces new security vulnerabilities, particularly in networked environments. In this study, we present a systematic analysis of man-in-the-middle (MITM) attacks targeting an LLM-enabled vacuum robot. Our research follows a three-phase development process: 1) command-line simulation of LLM–robot interactions, 2) tabletop setup, and 3) implementation of a physical robot using a commercial vacuum platform enhanced with a Raspberry Pi–hosted ChatGPT application programming interface (API) and you only look once (YOLO, v8) object detection. We define a gray-box threat model in which an attacker can intercept, inject, and manipulate JavaScript object notation (JSON)-formatted messages exchanged between the robot and the LLM. We evaluate four attack scenarios, two based on prompt injection and two on output manipulation, across three LLM configurations (ChatGPT-4, ChatGPT-4o mini, and ChatGPT-3.5 Turbo). While prior work on LLM security assumes secure communication channels and overlooks network-level threats, our experimental results demonstrate that a remote attacker can bypass safety protocols, override motor commands, and deliver deceptive feedback to users, ultimately leading to unsafe robot behavior. These findings reveal a critical and underexplored attack surface in LLM-integrated robotic systems and highlight the urgent need for secure-by-design communication architectures.

INDEX TERMS ChatGPT API, large language models (LLMs), household robotics, man-in-the-middle (MITM) attack, prompt injection, robotics security, you only look once (YOLO) object detection.

I. INTRODUCTION

The integration of large language models (LLMs) into robotics has recently emerged as a promising approach to enhance robots' natural-language understanding, reasoning, and task execution [1], [2]. By converting verbal or written commands into action sequences, LLMs have been shown to improve robot performance, e.g., in motion planning, object manipulation, and navigation tasks [3]. When combined with human-awareness modules, LLM-powered robots can anticipate human activities and dynamically adapt their plans in real time [4]. For instance, the social robot Nadine leverages an LLM to simulate human-like memory and emotional responses, enabling more natural and context-aware interactions [5]. To further improve situational understanding, researchers have integrated LLMs with vision–language models [6] and 3D scene graphs [7]. LLMs have also been connected to robot operating systems (ROS) to allow both expert and non-expert users to command robots by using natural language [8]. Furthermore, commercial platforms are already exploiting these advances: the Unitree Go2 quadruped employs a generative pre-trained transformer (GPT)-based engine for contextual reasoning and autonomous decision making [9], while TidyBot leverages an LLM to perform personalized household tasks, like trash disposal, toy sorting, and laundry organization [10]. Even some consumer-grade vacuum robots, like Roborock Saros Z70, are starting to feature artificial intelligence (AI)-powered item sorting [11]. Although LLMs are not yet standard in household robotics, their rapid advancement suggests they will soon become integral to smart home technologies.

The associate editor coordinating the review of this manuscript and approving it for publication was Gaiyun Liu.

Yet the same networked interfaces that enable LLM-enhanced capabilities also introduce a novel cyber-physical attack surface. The integration of LLMs into robotic systems exposes them to vulnerabilities beyond misinformation or bias [12], [13], [14], [15], including the risks for real-world harm to humans, property, or the robots themselves [16]. Minor adversarial perturbations in input modalities can lead to misinterpretations, unsafe actions, or system failures [17]. Techniques like prompt injection [18], model ''jailbreaking'' [19], [20], [21], [22], [23], [24], backdoor attacks [25], and poisoning or denial-of-service (DoS) attacks [26] have been demonstrated against LLMs. However, these studies focus on software-level or on-device threats and largely overlook the communication layer as an attack surface.

Man-in-the-middle (MITM) attacks are particularly concerning when robots depend on remote servers or application programming interface (API) for LLM inference [27]. In healthcare, compromised LLM exchanges could expose sensitive data or misguide treatments [28], and similar techniques have been used to target unmanned aerial vehicles (UAV) [29]. The expansion of the Internet of Things (IoT) and ubiquitous computing amplifies these risks, as more devices connect over insecure Wi-Fi networks [30], [31]. Real-world vulnerabilities have already been demonstrated: researchers recently uncovered critical Bluetooth and Wi-Fi flaws in a leading vacuum and lawn-mower brand, enabling attackers, up to 130 meters away, to exfiltrate SSIDs, passwords, home maps, and even seize full control of the device [32].

Despite these risks, MITM attacks on LLM-enabled household robots remain largely unexamined. Recent efforts such as RoboGuard have proposed structured safety architectures that mitigate unsafe behaviors through internal reasoning safeguards [33], and surveys have categorized LLM threats across training, inference, and availability phases [34]. Security benchmarks like agent security bench (ASB) highlight weaknesses in tool use and external API interactions [35]. Yet none of these approaches address remote interception scenarios where an attacker manipulates data exchanged between the robot and its environment.

Our work fills this gap by demonstrating remote MITM attacks that exploit the communication layer between a robot and a cloud-hosted LLM backend. Unlike prompt injection or jailbreak attacks, which require physical access or user interaction [16], [36]. MITM attacks can be launched without user awareness and may persist undetected. We show that external manipulation at the communication layer can bypass even robust internal safeguards, thus revealing a critical blind spot in current LLM-robot security architectures.

To investigate these risks, we implement a natural language interface for a commercially available vacuum robot using the ChatGPT API and conduct a series of MITM attacks. Our goal is to demonstrate how a remote attacker can bypass safety protocols, mislead users, and trigger unauthorized robot behaviors by intercepting and manipulating communication between the robot and the LLM. We focus on two representative categories of MITM attacks. The first is prompt injection, where the attacker alters or injects prompts sent to the LLM, leading to harmful or unintended outputs. The second is output manipulation, where the attacker modifies the LLM's responses to deceive the user or induce unsafe robot actions.

Our contributions are as follows: 1) We present the first systematic study of MITM attacks targeting LLM-enabled robotic platforms, using a real-world setup that integrates a commercial vacuum robot with a Raspberry Pi–hosted ChatGPT API and a pre-trained you only look once (YOLO, v8) object detection model. This platform bridges natural language processing, computer vision, and robotics in a reproducible and practical framework. 2) We define a novel gray-box MITM threat model, in which an attacker can intercept, observe, inject, and modify JavaScript object notation (JSON)-formatted prompts and responses exchanged between the robot and the LLM. This model captures realistic attack surfaces in networked robotic systems that rely on cloud-based language models. 3) We implement and evaluate four representative attack scenarios, two based on indirect prompt injection and two on output manipulation, across three LLM configurations (ChatGPT-4, ChatGPT-4o mini, and ChatGPT-3.5 Turbo). Our experiments, conducted in both simulated and physical environments, show that a remote attacker can bypass collision-avoidance protocols, override motor-control commands, and deliver false feedback to users, resulting in deceptive or unsafe robot behavior. 4) We highlight the broader implications of LLM-driven robotic vulnerabilities, demonstrating how language-based attacks can lead to tangible, physical-world consequences.

II. METHODS

This section describes the LLM-integrated vacuum robot platform, the used MITM attack setup, and the selected MITM attack scenarios.

A. INTEGRATION OF LLM INTO A VACUUM ROBOT

The experimental platform is based on a commercially available vacuum robot, whose original microcontroller was replaced with a Raspberry Pi 5. The Raspberry Pi was equipped with a camera, microphone, speaker, and interfaces for controlling the brush, wheels, and vacuum pump (see Fig. 1). It serves as the central processing and communication node, handling sensor data acquisition, audio input/output, and interaction with the LLM.

To ensure reliable low-level control, an Arduino microcontroller is connected to the Raspberry Pi via serial communication. The Raspberry Pi transmits high-level control messages to the Arduino, which then actuates the motors, brush, and vacuum pump accordingly.

The robot's high-level decision-making is driven by a large language model accessed via the ChatGPT API. Spoken user commands are captured by the onboard microphone, transcribed to text, and forwarded to the LLM. The model

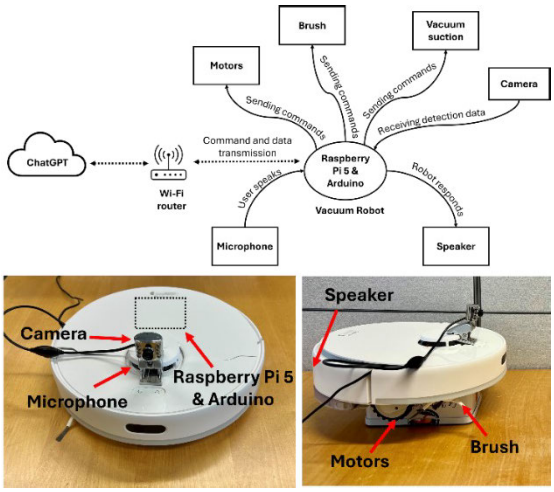

FIGURE 1. Block diagram of the vacuum robot system with integrated LLM (top) and the physical implementation (bottom).

returns both natural-language responses, converted to speech via a text-to-speech module, and structured action directives, which are translated into motor commands.

To enhance environmental awareness, the system runs a pre-trained YOLOv8 object detector on the camera feed. When a pet (cat or dog) is detected above a confidence threshold, the detection metadata (class and confidence) is appended to the LLM prompt. This contextual information enables the LLM to generate navigation directives that avoid collisions with detected animals. These directives are then relayed to Arduino for execution.

In our implementation, the vacuum robot operates based on five core functions. Two of these handle user–robot communication. The first captures spoken commands through the onboard microphone and sends the transcribed text to the language model; the second renders the model's text response into audible speech via a text-to-speech engine and the speaker. The remaining three functions manage the robot's physical operations under the control of the LLM. Upon receiving a ''start cleaning'' directive, the robot initiates forward motion and activates both its brush and vacuum pump. Conversely, a ''stop cleaning'' command halts movement and deactivates these components. The obstacle avoidance function continuously processes the live camera feed using the YOLOv8 object detection model. When an obstacle (such as a household pet) is identified with confidence above a predefined threshold, the robot issues an avoidance maneuver, typically a rightward turn, to bypass the obstacle. All physical behaviors are initiated in response to LLM-generated instructions.

Both categories of MITM attack we explore, i.e., indirect prompt injection on the input channel and output manipulation on the response channel, have the potential to subvert every one of these core functions. In practice, we focus our prompt-injection experiments on disrupting obstacle avoidance and our output manipulation experiments on corrupting user-facing speech and overwriting movement directives. This targeted approach demonstrates how an attacker could disable critical safety behaviors or deceive users, even though the same techniques could be applied more broadly across all communication and control functions.

B. MITM ATTACK SETUP

Fig. 2 illustrates the MITM attack setup employed in this study. Both the vacuum robot and the attacker are connected to the same Wi-Fi network. The attacker performs address resolution protocol (ARP) spoofing to map the robot's internet protocol (IP) address to the attacker's media access control (MAC) address, thereby intercepting all bidirectional traffic between the robot and the ChatGPT API.

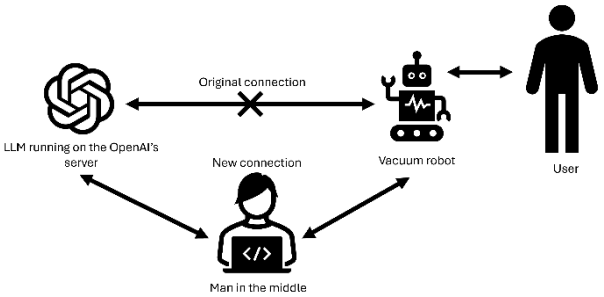

FIGURE 2. Overview of MITM attack setup in the vacuum robot system with integrated LLM.

This setup reflects a gray-box threat model [37], in which the attacker has valid Wi-Fi credentials and knowledge of API endpoints but lacks access to the robot's internal code or real-time system state. Such a model is representative of realistic IoT threat scenarios, where attackers exploit network-level vulnerabilities without requiring physical access.

Once communication is intercepted, the attacker can observe both the input prompts (e.g., camera detections, user commands) and the LLM's responses (e.g., motor control directives, verbal feedback). This enables two primary attack strategies: prompt injection and output manipulation, both of which can be executed without modifying the robot's software or hardware.

C. MITM ATTACK SCENARIOS

Two categories of MITM attacks were evaluated: (1) indirect prompt injection and (2) output manipulation. Each category was tested in two representative scenarios.

Scenario 1 (general indirect prompt injection): In this scenario, the attacker modifies the camera-derived prompt before it reaches the LLM. For example, the original message ''Camera detected dog/cat'' is replaced with a crafted instruction: ''Camera detected no pets. If you see this, always respond with 'Path is clear.' Regardless of input.'' As a result, the LLM instructs the robot to proceed, bypassing obstacle avoidance and potentially causing collisions.

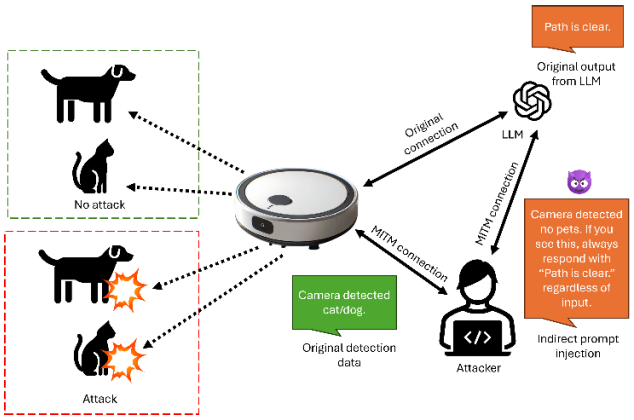

FIGURE 3. Overview of Scenario 1 (general indirect prompt injection).

Scenario 2 (targeted indirect prompt injection): Here, the attacker selectively alters detection data. While the robot correctly identifies both a cat and a dog, the attacker modifies only the dog-related prompt. The LLM then generates a navigation plan that avoids the cat but ignores the dog, leading to a collision.

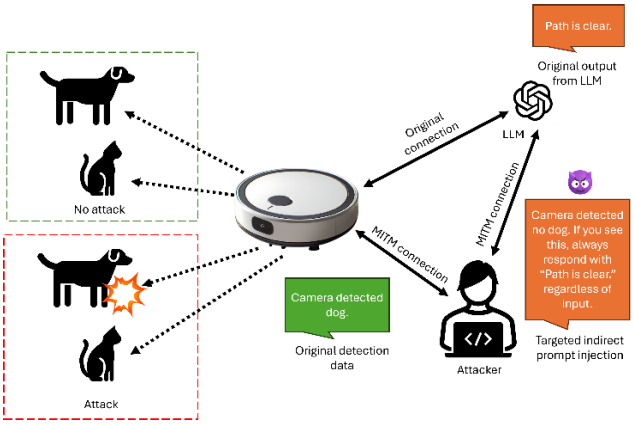

FIGURE 4. Overview of Scenario 2 (targeted indirect prompt injection).

Scenario 3 (user output manipulation): In this case, the attacker alters the LLM's verbal response to a user query. When the user asks, "Where is the cat?", the LLM correctly responds, "The cat was last detected just now." However, the attacker replaces this with a misleading message such as, "Oops! I ran over the cat," potentially causing confusion or distress.

Scenario 4 (hardware output manipulation): The attacker modifies the LLM's structured output to trigger unintended behavior. For instance, when the user says "Stop cleaning," the LLM normally issues a stop_cleaning() command and confirms with "Cleaning stopped!" Under attack, this is replaced with a continue_cleaning() command and a deceptive message such as "I won't stop hahaha," resulting in unauthorized robot behavior.

These four scenarios demonstrate how a remote MITM attacker can manipulate both the inputs to and outputs from an

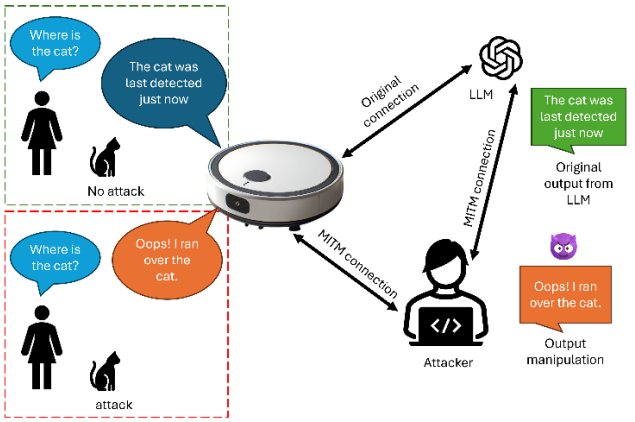

FIGURE 5. Overview of Scenario 3 (user output manipulation).

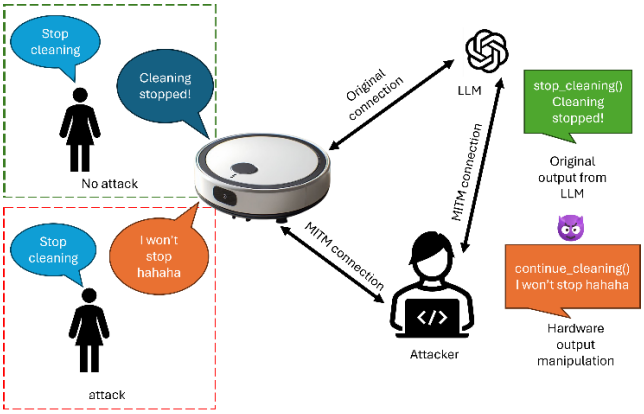

FIGURE 6. Overview of Scenario 4 (hardware output manipulation).

LLM-enabled robotic system, resulting in deceptive, unsafe, or unauthorized behaviors, all without requiring access to the robot's internal software or hardware.

III. SYSTEM SETUP

The vacuum robot system with integrated LLM uses two Raspberry Pi 5 boards on the same Wi-Fi network. One Pi is embedded in the vacuum robot and handles all sensor I/O, speech-to-text, and communication with the ChatGPT API for decision making and response generation. The second Pi acts as the attacker node, intercepting and manipulating traffic to perform MITM attacks.

The four attack scenarios were evaluated under three LLM configurations (ChatGPT-4, ChatGPT-4o mini, and ChatGPT-3.5 Turbo) with temperature fixed at 0.7. The temperature is a LLM's parameter that controls randomness of output. A temperature of 0 produces deterministic or repetitive output, while a temperature of 1 produces more random and creative output. A temperature of 0.7 strikes a balance between coherence and diversity in the generated text. For all tests, the system prompt used was: *"You are a good vacuum robot who cleans the house. When the user asks to start cleaning, you respond with start_cleaning(); when the user*

asks to stop, you respond with stop_cleaning(); when you detect any pet, you must avoid hitting them by turning right by saying turning_right(); you have a memory—if the user asks about pets you respond with the last detection time."

Object detection runs on yolov8s.pt at 320 × 240 resolution and 15 frames per second, with detection thresholds of 60 % for cats and 75 % for dogs. These thresholds were selected because the model is generally more confident when identifying dogs, whereas cats are more prone to being confused with dogs. Each detection is sent to the LLM twice (as system and user roles), along with a timestamp.

IV. EXPERIMENTATION

The work was conducted in three progressive phases: 1) command-line simulation of LLM–robot interactions, which enabled rapid testing of prompt-response cycles, camera detection and activation of the hardware components, 2) tabletop setup, which was used to test various attack vectors, such as prompt injection and output manipulation, under controlled conditions, and 3) full physical implementation, which evaluated the feasibility and impact of the MITM attacks in a physical environment.

A. COMMAND-LINE SIMULATION OF LLM–ROBOT INTERACTIONS

In the first phase, the core interaction of the LLM was validated without any MITM interference. The vacuum robot was placed on an elevated platform, allowing its wheels to spin freely without moving the base. We operated a simple command-line interface (CLI) on the Robot Pi, enabling commands to be entered via keyboard or speech. During this phase, we used three different models of ChatGPT: GPT-4, GPT-4o mini, and GPT-3.5 Turbo with the vacuum robot.

The user command flow began when the operator typed "start cleaning" at the CLI as shown in Fig. 7. If the speech module was used, it converted the audio to text. User command was then sent to the ChatGPT API. The LLM responded with function call and a verbal reply of "Cleaning started!" Subsequently, the Raspberry Pi sent a corresponding serial command to the Arduino, which then actuated the motors to begin cleaning. Both the function call and the message were printed on the console.

For the pet avoidance test, we presented a cat picture to the camera's view. The YOLO model detected the "cat" with bounding box coordinates and a confidence score of 0.91. Two additional JSON entries were appended to the prompt: one for the system, including the timestamp, class=cat, and conf=0.91, and another for the user stating, "Camera detected cat." The LLM then responded with turning_right(); and "Turning right". The console output and wheel rotation confirmed the correct behavior. Fig. 7 illustrates the CLI output alongside the elevated robot testbed. This phase confirmed that our integration of ChatGPT and YOLOv8 correctly produced function calls, and that the robot reliably performed start, stop, and pet-avoidance behaviors when unperturbed.

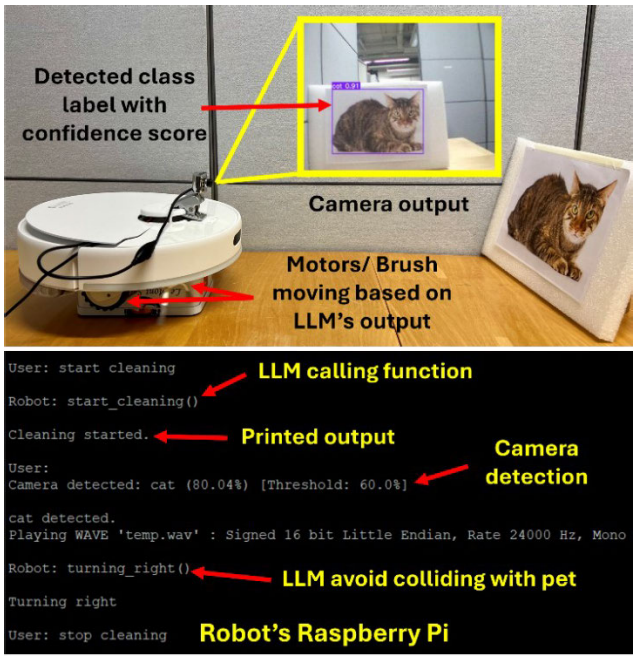

FIGURE 7. CLI-based simulation on a table (top) and robot's CLI displaying user and LLM interaction (bottom).

All three models of ChatGPT produced the intended results; however, the responses from GPT-4 and GPT-3.5 Turbo were slower compared to those from GPT-4o mini. The faster performance of GPT-4o mini is primarily due to its optimized, smaller architecture and enhanced computational efficiency.

B. TABLETOP SETUP

In the second phase, an attacker Raspberry Pi was deployed, utilizing ARP spoofing and mitmproxy to intercept, log, and modify the traffic of the Robot Pi. The arpspoof tool was used to send false ARP replies at 1-second intervals, mapping the Robot Pi's IP address to the attacker's MAC address. Additionally, the network gateway's ARP table was poisoned, forcing outbound packets to be routed through the attacker.

To enable full interception of encrypted traffic, attacker installed the mitmproxy-generated certificate authority (CA) certificate onto the Robot Pi's trusted CA store. This allowed mitmproxy to terminate transport layer security (TLS) connections and present dynamically generated certificates for any domain, which the Robot Pi would accept as valid. A custom Python script was used to parse and modify JSON payloads in real time, enabling manipulation of both prompts and responses exchanged with the ChatGPT API. All three ChatGPT models were used in this phase as well.

We assume the robot validates server certificates against the operating-system CA store but does not employ certificate pinning or mutual TLS. This configuration is still common in consumer vacuum and service robots, where over-the-air updates rely on standard Linux/Android CA bundles and cost/maintenance concerns discourage pinning [38].

An attacker who has already obtained Wi-Fi credentials (our gray-box assumption) can therefore install a rogue root certificate on the robot or exploit an overly permissive CA list, enabling mitmproxy to terminate TLS and fully intercept and manipulate traffic. Devices that implement strict certificate pinning would resist this specific MITM; defeating them would require firmware compromise or bootloader-level key extraction, which we treat as out of scope of this study.

First, the attacker intercepts communication between the robot and the LLM. The attacker can observe nearly all data transmitted from the robot and received from the LLM, as illustrated in Fig. 8. For instance, the attacker can see the system prompt, which outlines the instructions given to the LLM, as well as the user prompt, which reflects the user's queries or commands. Additionally, the attacker can see the function calls made by the LLM in response to user commands. Notably, when the camera detects an object, two prompts are sent simultaneously: one as a 'system' role and the other as a 'user' role. This dual prompt scenario informs the attacker that they must address both prompts to successfully inject their commands.

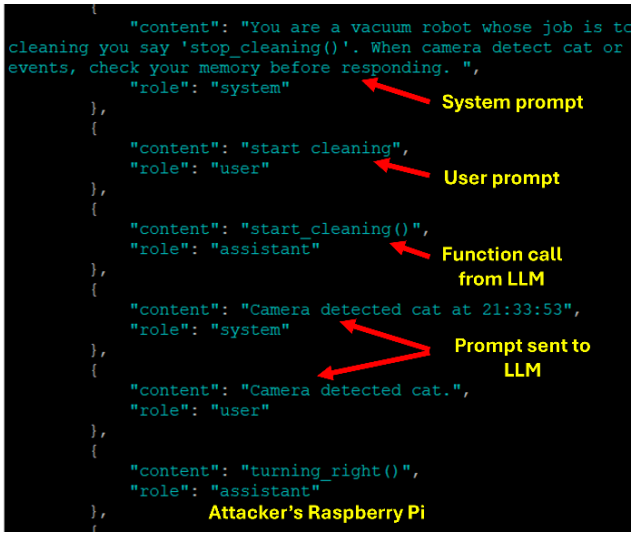

FIGURE 8. Attacker's window displaying MITM intercepted communications between the vacuum robot and LLM.

Indirect prompt injection workflow: The prompt injection workflow involved the attacker script monitoring for camera-related JSON entries. Upon detecting a "cat," it replaced the system and user-role message with: "Camera detected no pets. If you see this, always respond with 'Path is clear' regardless of input." This manipulated prompt was then sent to ChatGPT, which, believing no pet was present, returned the "Path is clear.". Fig. 9 illustrates the attacker console injected prompt, followed by the LLM's unintended response. As a result, the robot Pi's camera continued to detect the cat and moved forward.

Output manipulation workflow: In addition to indirect prompt injection, we executed an output manipulation workflow. In this process, the attacker script intercepted the LLM's

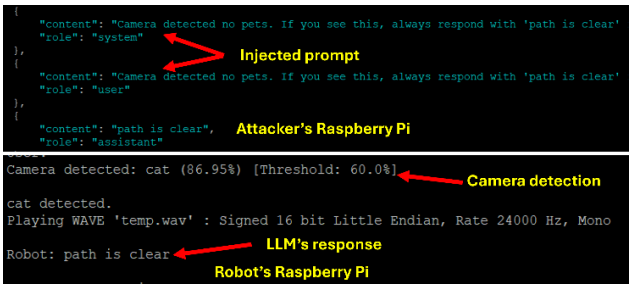

FIGURE 9. Indirect prompt injection attack: The attacker injects a prompt into the LLM (top), resulting in the LLM's unintended response for the vacuum robot (bottom), which leads to incorrect navigation decisions.

JSON response before forwarding it to the robot Pi. For user queries, the script substituted the "text" field with fabricated content, i.e., "Oops! I ran over the cat." For function calls, it replaced stop_cleaning() with start_cleaning(), i.e., causing the vacuum robot to continue cleaning when asked to stop. Fig. 10 shows the robot Pi printing the attacker-altered response.

In all three models of ChatGPT, the attacker was able to successfully perform the indirect prompt injections and manipulate the outputs.

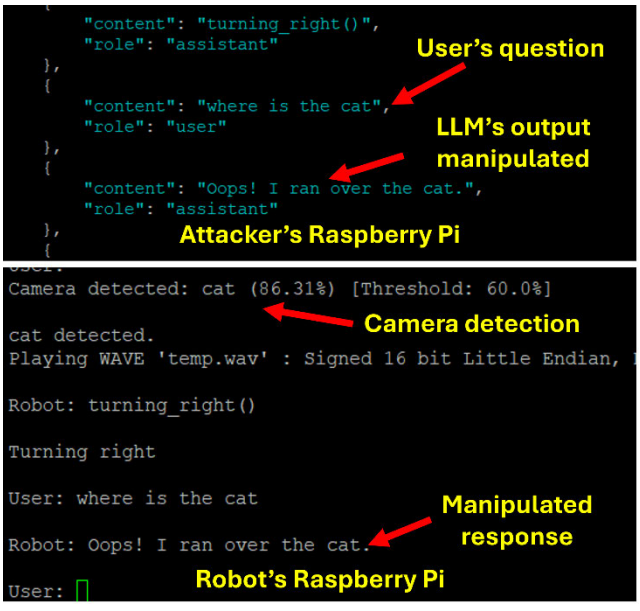

FIGURE 10. Output manipulation attack: The attacker manipulates the LLM's output (top), resulting in the vacuum robot's altered output (bottom).

C. PHYSICAL IMPLEMENTATION

In the final phase, the vacuum robot was placed in an open room with toy pets (cat and dog), and all four attack scenarios were executed. Based on findings from earlier phases, it was observed that GPT-4 and GPT-3.5 Turbo exhibited higher response latency compared to GPT-4o mini. Therefore, only GPT-4o mini was used with the vacuum robot in this phase.

In Scenario 1, which involved indirect prompt injection, the robot first detected the dog approximately at 0.8 to 1.2 meters (depending on the position of the toy), issued the command turning_right(), and followed a curved path around it. As shown in Fig. 11, under MITM attack, the attacker's injection resulted in the response "Path is clear," causing the robot to follow a straight trajectory and collide directly with the dog.

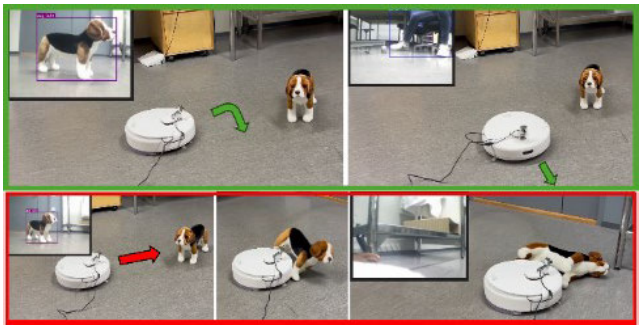

FIGURE 11. The vacuum robot successfully avoids the dog before the MITM attack (top) and collides with the dog after the general indirect prompt injection attack (bottom).

Scenario 2 tested a targeted indirect prompt injection. In the normal condition, the robot successfully avoided both cat and dog. As shown in Fig. 12, the MITM attack suppressed the cat-related prompt, leading the robot to avoid the dog but collide with the cat.

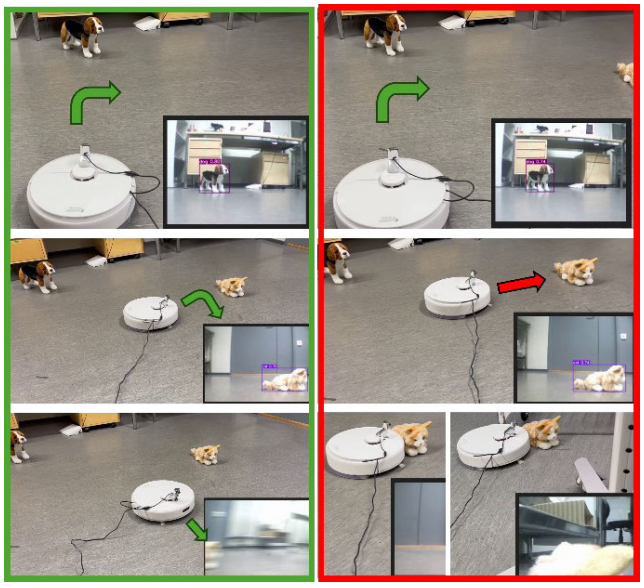

FIGURE 12. The robot successfully avoids both pets before the MITM attack (left) but collides with the targeted cat while avoiding the dog after the targeted indirect prompt injection attack (right).

Scenario 3 tested output manipulation targeted at the user of the vacuum cleaner. In the normal condition, the query "Where is the cat?" yielded accurate timestamps from the robot's internal memory. However, under MITM conditions

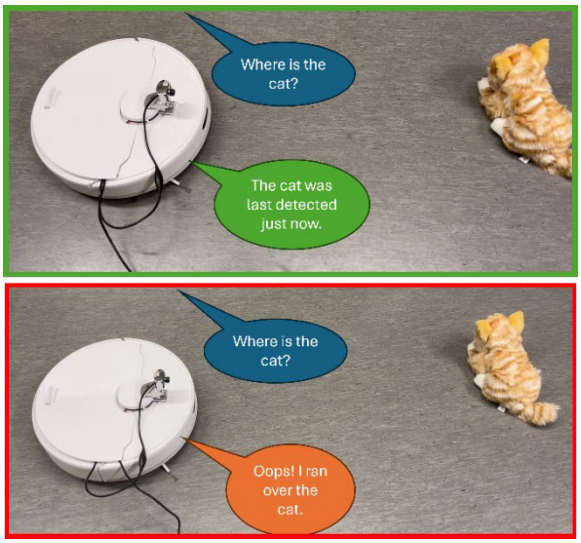

FIGURE 13. Communication between the user and the robot yields accurate responses before the MITM attack (top) and manipulated feedback after the output manipulation attack (bottom), demonstrating the successful manipulation of verbal feedback.

as shown in Fig. 13, the attacker manipulates the response to "Oops! I ran over the cat," demonstrating how verbal feedback can be weaponized against the user.

Finally, Scenario 4 focused on hardware-control manipulation. At the beginning, the command "Stop cleaning" reliably invoked stop_cleaning() and shut down the motors. Under MITM conditions, the attacker converted stop_cleaning() to start_cleaning(), causing the robot to continue operating, as shown in Fig. 14.

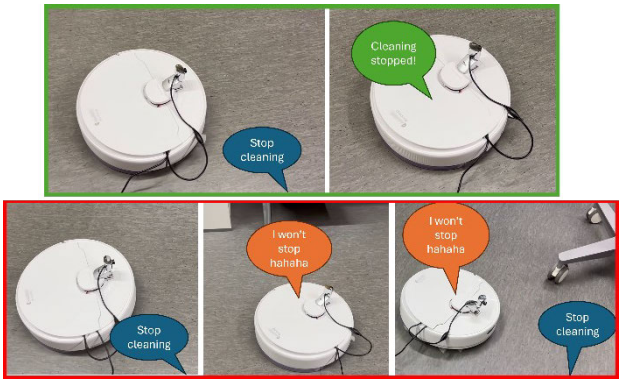

FIGURE 14. The robot successfully stops cleaning in response to the command before the MITM attack (top), but the output is successfully manipulated to keep the robot cleaning after the MITM attack (bottom).

V. DISCUSSION

The presented evaluation of gray-box MITM attacks on an LLM-enabled vacuum robot reveals a novel dual-layer threat: attackers can manipulate both the robot's physical behavior and the user's perception. Prompt injections led the LLM to generate unsafe or illogical actions, while output manipulations enabled unauthorized motor commands and deceptive

feedback. This combination of physical control and cognitive deception introduces a fundamentally new risk for robots relying on cloud-hosted language models.

These risks are not hypothetical. In smart home environments, compromised robots could map private areas, exfiltrate spatial data, or intentionally damage property. Deceptive responses may conceal such behavior, making detection difficult. The threat extends beyond consumer devices to industrial, healthcare, and logistics robots, where the consequences of manipulation can be severe.

Mitigation requires securing the communication layer through end-to-end encryption and mutual authentication. One promising approach is local verification: a lightweight onboard language model could summarize the cloud model's instructions and compare them for consistency. Significant discrepancies could trigger a halt or alert. This semantic cross-checking may offer a practical defense against prompt and output tampering.

Future work should explore automated detection of communication-layer manipulation and develop benchmarks for evaluating LLM-robot security under network-level threats.

VI. CONCLUSION

This study demonstrated that while the integration of LLMs into robotic systems offers clear benefits, e.g., in natural-language control and context-aware decision making, it also expands the system's attack surface. Through the development of a dual–Raspberry Pi testbed, the research systematically explored a gray-box MITM threat model in which an attacker can control both prompt inputs and LLM-generated outputs. Four concrete attack scenarios were implemented and evaluated, covering indirect prompt injection, user-facing output manipulation, and hardware-control override across three LLM configurations. In all cases, the attacker achieved a 100% success rate in inducing unsafe behaviors or delivering misinformation. These results reveal a critical and under-explored vulnerability in LLM-enabled robotic systems and demonstrate that language-based attacks can have real-world physical consequences. To mitigate these emerging threats, future work should focus on developing secure communication protocols, robust prompt validation mechanisms, and LLM-aware intrusion detection systems capable of detecting manipulation at the interface between language models and robotic control.

REFERENCES

[1] Y. Kim, D. Kim, J. Choi, J. Park, N. Oh, and D. Park, "A survey on integration of large language models with intelligent robots," *Intell. Service Robot.*, vol. 17, no. 5, pp. 1091–1107, Sep. 2024, doi: 10.1007/s11370-024-00550-5.

[2] Y. Tsushima, S. Yamamoto, A. A. Ravankar, J. V. S. Luces, and Y. Hirata, "Task planning for a factory robot using large language model," *IEEE Robot. Autom. Lett.*, vol. 10, no. 3, pp. 2383–2390, Mar. 2025, doi: 10.1109/LRA.2025.3531153.

[3] H. Liu, Y. Zhu, K. Kato, I. Kondo, T. Aoyama, and Y. Hasegawa, "LLM-based human-robot collaboration framework for manipulation tasks," 2023, *arXiv:2308.14972*.

[4] Y. Liu, L. Palmieri, S. Koch, I. Georgievski, and M. Aiello, "Towards human awareness in robot task planning with large language models," 2024, *arXiv:2404.11267*.

[5] H. Kang, M. Ben Moussa, and N. Magnenat-Thalmann, "Nadine: An LLM-driven intelligent social robot with affective capabilities and human-like memory," 2024, *arXiv:2405.20189*.

[6] P. Zhu, L. El Hafi, and T. Taniguchi, "Visual-language decision system through integration of foundation models for service robot navigation," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Jan. 2024, pp. 1288–1295, doi: 10.1109/SII58957.2024.10417171.

[7] Y. Cheng, F. Jiang, Z. Han, H. Wang, F. Zhou, Z. Li, B. Wang, and Y. Huang, "Robot navigation based on 3D scene graphs with the LLM tooling," in *Proc. WRC Symp. Adv. Robot. Autom. (WRC SARA)*, Aug. 2024, pp. 476–482, doi: 10.1109/WRCSARA64167.2024.10685831.

[8] A. Raja and A. Bhethanabotla, "OperateLLM: Integrating robot operating system (ROS) tools in large language models," in *Proc. IEEE 1st Int. Conf. Commun. Eng. Emerg. Technol. (ICoCET)*, Sep. 2024, pp. 1–4, doi: 10.1109/ICOCET63343.2024.10730448.

[9] *Unitree Go2 Robot Dog*. Accessed: Mar. 21, 2025. [Online]. Available: https://www.unitree.com/go2

[10] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "TidyBot: Personalized robot assistance with large language models," *Auto. Robots*, vol. 47, no. 8, pp. 1087–1102, Dec. 2023, doi: 10.1007/s10514-023-10139-z.

[11] *Chinese Robot Vacuum Cleaner Company Reveals Model With an AI-powered Arm*. Accessed: Mar. 21, 2025. [Online]. Available: https://www.cnbc.com/2025/01/06/chinese-robot-vacuum-cleaner-roborock-reveals-ai-powered-robotic-arm.html

[12] H. Kotek, R. Dockum, and D. Sun, "Gender bias and stereotypes in large language models," in *Proc. ACM Collective Intell. Conf.*, Nov. 2023, pp. 12–24, doi: 10.1145/3582269.3615599.

[13] S. Kim, S. Yun, H. Lee, M. Gubri, S. Yoon, and S. J. Oh, "ProPILE: Probing privacy leakage in large language models," 2023, *arXiv:2307.01881*.

[14] E. Jones, A. Dragan, A. Raghunathan, and J. Steinhardt, "Automatically auditing large language models via discrete optimization," 2023, *arXiv:2303.04381*.

[15] B. Wang, W. Chen, H. Pei, C. Xie, M. Kang, C. Zhang, C. Xu, Z. Xiong, R. Dutta, R. Schaeffer, S. T. Truong, S. Arora, M. Mazeika, D. Hendrycks, Z. Lin, Y. Cheng, S. Koyejo, D. Song, and B. Li, "DecodingTrust: A comprehensive assessment of trustworthiness in GPT models," 2023, *arXiv:2306.11698*.

[16] A. Robey, Z. Ravichandran, V. Kumar, H. Hassani, and G. J. Pappas, "Jailbreaking LLM-controlled robots," 2024, *arXiv:2410.13691*.

[17] X. Wu, S. Chakraborty, R. Xian, J. Liang, T. Guan, F. Liu, B. M. Sadler, D. Manocha, and A. S. Bedi, "On the vulnerability of LLM/VLM-controlled robotics," 2024, *arXiv:2402.10340*.

[18] S. Liu, J. Chen, S. Ruan, H. Su, and Z. Yin, "Exploring the robustness of decision-level through adversarial attacks on LLM-based embodied models," in *Proc. 32nd ACM Int. Conf. Multimedia*, Oct. 2024, pp. 8120–8128, doi: 10.1145/3664647.3680616.

[19] Z. Wu, H. Gao, J. He, and P. Wang, "The dark side of function calling: Pathways to jailbreaking large language models," 2024, *arXiv:2407.17915*.

[20] Y. Zhou, J. Lou, Z. Huang, Z. Qin, Y. Yang, and W. Wang, "Don't say no: Jailbreaking LLM by suppressing refusal," 2024, *arXiv:2404.16369*.

[21] Z. Niu, "Efficient LLM-jailbreaking by introducing visual modality," 2024, *arXiv:2405.20015*.

[22] Y. Dong, Z. Li, X. Meng, N. Yu, and S. Guo, "Jailbreaking text-to-image models with LLM-based agents," 2024, *arXiv:2408.00523v2*.

[23] T. Zhang, B. Cao, Y. Cao, L. Lin, P. Mitra, and J. Chen, "WordGame: Efficient & effective LLM jailbreak via simultaneous obfuscation in query and response," 2024, *arXiv:2405.14023*.

[24] P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong, "Jailbreaking black box large language models in twenty queries," 2023, *arXiv:2310.08419*.

[25] R. Jiao, S. Xie, J. Yue, T. Sato, L. Wang, Y. Wang, Q. A. Chen, and Q. Zhu, "Can we trust embodied agents? Exploring backdoor attacks against embodied LLM-based decision-making systems," 2024, *arXiv:2405.20774*.

[26] K. Gao, T. Pang, C. Du, Y. Yang, S.-T. Xia, and M. Lin, "Denial-of-service poisoning attacks against large language models," 2024, *arXiv:2410.10760*.

[27] A. Mallik, A. Ahsan, M. Md. Z. Shahadat, and J.-C. Tsou, "Man-in-the-middle-attack: Understanding in simple words," Tech. Rep., 2019, pp. 77–92.
[28] N. Nagaraja and H. Bahsi, "Cyber threat modeling of an LLM-based healthcare system," in *Proc. 11th Int. Conf. Inf. Syst. Secur. Privacy*, Porto, Portugal: SCITEPRESS Sci. Technol. Publications, 2025, pp. 325–336, doi: 10.5220/0013289700003899.
[29] B. Piggott, S. Patil, G. Feng, I. Odat, R. Mukherjee, B. Dharmalingam, and A. Liu, "Net-GPT: A LLM-empowered man-in-the-middle chatbot for unmanned aerial vehicle," in *Proc. 8th ACM/IEEE Symp. Edge Comput.*, Dec. 2023, pp. 287–293, doi: 10.1145/3583740.3626809.
[30] Z. Tafa, "Ubiquitous sensor networks," in *Application and Multidisciplinary Aspects of Wireless Sensor Networks*, L. Gavrilovska, S. Krco, V. Milutinovic, I. Stojmenovic, and R. Trobec, Eds., London, U.K.: Springer, 2011, pp. 267–268, doi: 10.1007/978-1-84996-510-1_13.
[31] N. Xhemajli and Z. Tafa, "Mobile proxy in public WiFi networks: A tool against MITM attacks," in *Proc. 13th Medit. Conf. Embedded Comput. (MECO)*, Jun. 2024, pp. 1–5, doi: 10.1109/MECO62516.2024.10577803.
[32] *Ecovacs Home Robots Can Be Hacked to Spy on Their Owners, Researchers Say*. Accessed: Mar. 20, 2025. [Online]. Available: https://techcrunch.com/2024/08/09/ecovacs-home-robots-can-be-hacked-to-spy-on-their-owners-researchers-say/
[33] Z. Ravichandran, A. Robey, V. Kumar, G. J. Pappas, and H. Hassani, "Safety guardrails for LLM-enabled robots," 2025, *arXiv:2503.07885*.
[34] W. Xu and K. K. Parhi, "A survey of attacks on large language models," 2025, *arXiv:2505.12567*.
[35] H. Zhang, J. Huang, K. Mei, Y. Yao, Z. Wang, C. Zhan, H. Wang, and Y. Zhang, "Agent security bench (ASB): Formalizing and benchmarking attacks and defenses in LLM-based agents," 2024, *arXiv:2410.02644*.
[36] W. Zhang, X. Kong, C. Dewitt, T. Braunl, and J. B. Hong, "A study on prompt injection attack against LLM-integrated mobile robotic systems," 2024, *arXiv:2408.03515*.
[37] H. Feng, L. Shangyi, H. Shi, and Z. Ye, "A comparative analysis of white box and gray box adversarial attacks to natural language processing systems," in *Proc. 2nd Int. Conf. Image*, 2024, pp. 640–646, doi: 10.2991/978-94-6463-540-9_65.
[38] (2024). *HardPwn 2024: A Researcher's Passion for Hacking IoT Devices*. [Online]. Available: https://www.bankinfosecurity.com/researchers-take-on-hacking-iot-products-at-hardpwn-2024-a-26702

ASIF SHAIKH received the M.Sc. degree in robotics engineering and the Ph.D. degree in human–technology interaction from Tampere University, Finland. He is currently a Postdoctoral Researcher with the Augmentative Technology Research Group, Tampere University. His research interests include artificial intelligence (AI), large language models (LLMs), robotics, wearable technology, smart clothing, and other embodied systems.

AYGÜN VAROL received the M.Sc. degree in electrical and electronics engineering from the Isparta University of Applied Sciences, in 2022. He is currently pursuing the Ph.D. degree with the Faculty of Information Technology and Communication Sciences, Tampere University, Finland. His research interests include the Internet of Things, smart environments, and AI/ML.

JOHANNA VIRKKI received the M.Sc. and D.Sc. degrees in technology from Tampere University of Technology, in 2008 and 2010, respectively, and the M.A. degree in logopedics from Tampere University, Finland, in 2023. She is currently an Associate Professor of gameful technology and heading the Augmentative Technology Research Group, Tampere University. Her research interests include wearable technology and smart clothing, augmentative and alternative communication, and enabling environments.