應用 SDN 實現針對 AI 生成惡意流量之自適應防禦架構

Adaptive Defense Architecture for AI-Generated Malicious Traffic Based on SDN

Abstract

With the advancement of AI technology, cyber attackers are increasingly leveraging generative AI to automate social engineering, malware creation, phishing attacks, and adversarial threats. Traditional security measures struggle to keep up with these evolving attack vectors, necessitating adaptive and intelligent defense mechanisms. Software-Defined Networking (SDN) offers a dynamic approach to traffic monitoring and anomaly detection, making it a promising solution against AI-generated cyber threats.

This study explores the security risks posed by generative AI, focusing on its applications in cyberattacks such as deepfake-driven fraud, AI-enhanced phishing, adversarial machine learning attacks, and automated malware obfuscation. In response, we propose an SDN-based adaptive defense architecture capable of detecting and mitigating AI-generated malicious traffic. By integrating centralized control, anomaly detection, and AI-enhanced security policies, SDN can dynamically analyze and respond to emerging threats in real time. This research highlights the necessity of an AI-aware cybersecurity framework that leverages SDN for proactive defense, ensuring network resilience against the rapid evolution of AI-driven cyber threats.

Keywords: Software-Defined Networking (SDN), Generative AI, Cybersecurity, Anomaly detection, Adaptive defense, AI-generated traffic, Adversarial machine learning, Network resilience.

Introduction

The rapid advancement of artificial intelligence (AI) has revolutionized various industries,

providing unprecedented efficiency and automation. However, this progress has also introduced new security challenges [1, 2]. In recent years, cybercriminals have begun leveraging generative AI models to automate and enhance their attacks, making them more sophisticated and difficult to detect [1, 3]. From AI-generated phishing emails and deepfake scams to automated malware creation and adversarial attacks against machine learning systems, the cybersecurity landscape is facing an evolving set of AI-driven threats [1, 3, 4].

One notable incident underscoring this growing threat is a recent cyberattack in which a group of high school students exploited generative AI tools to breach a major telecommunications provider. Using AI-assisted scripting, they successfully infiltrated the system, stole hundreds of thousands of user credentials, and fraudulently activated SIM cards for illicit profit [5]. This case highlights the increasing accessibility of AI-powered hacking techniques and the urgent need for adaptive cybersecurity solutions [2, 6].

Traditional network security mechanisms struggle to cope with these emerging threats due to their reactive nature and static rule-based approaches [2]. To address this, Software-Defined Networking (SDN) presents a promising alternative by enabling dynamic traffic analysis, centralized control, and intelligent anomaly detection [4, 6, 7]. SDN's ability to adaptively respond to network threats in real time makes it an ideal foundation for mitigating AI-generated cyberattacks [4, 8].

This study aims to analyze the role of generative AI in modern cyber threats and propose an SDN-based adaptive defense architecture to counter AI-generated malicious traffic [4, 7]. By integrating AI-enhanced anomaly detection techniques with SDN's programmable framework, we seek to develop a proactive defense mechanism capable of detecting, analyzing, and mitigating evolving AI-driven cyber threats [6, 8].

1.1 Generative AI (GenAI) in Cybersecurity Attacks

Generative AI (GenAI) is no longer confined to creative domains such as text or image generation; it is increasingly being exploited as a tool for launching sophisticated cyberattacks [1, 3]. Malicious actors are leveraging GenAI to automate and enhance various attack vectors—crafting highly convincing phishing emails, generating deepfake videos and audio, writing obfuscated malware code, and even launching social engineering campaigns with unprecedented realism [3, 8].

By utilizing Large Language Models (LLMs) like ChatGPT or open-source alternatives,

attackers can generate convincing natural language content to manipulate users or automate parts of the hacking process [3, 5]. This democratization of advanced attack capabilities has significantly lowered the barrier to entry for cybercriminals, allowing even low-skill actors to execute complex threats [3, 5].

In this context, understanding how GenAI is being weaponized for cyberattacks is crucial. The dynamic and deceptive nature of GenAI-generated content presents unique challenges to traditional detection systems, making it necessary to explore adaptive, intelligence-driven security solutions that can detect and counter AI-powered threats effectively [2, 3, 7].

1.2 Software-Defined Networking (SDN) in Cybersecurity Defense

As AI-powered threats evolve in complexity and speed, traditional static security measures—such as rule-based firewalls and intrusion detection systems—often fall short [2, 6]. Software-Defined Networking (SDN) offers a promising alternative by enabling real-time adaptability and centralized control in network defense strategies [4, 6, 7].

SDN's architecture decouples the control plane from the data plane, allowing a central controller to dynamically manage network behavior. This centralization is particularly advantageous for cybersecurity, as it facilitates real-time traffic monitoring, rapid policy enforcement, and seamless integration with AI-driven anomaly detection systems [3, 6].

In practical applications, SDN can detect and respond to malicious traffic patterns—such as those generated by GenAI-driven attacks—by dynamically rerouting traffic, isolating compromised segments, or deploying targeted countermeasures [4, 7]. Its programmable nature allows for the implementation of proactive security mechanisms that evolve alongside emerging threats [4, 6, 8].

This study focuses on leveraging the programmability of SDN in conjunction with AI-enhanced detection methods to build an adaptive defense architecture. The goal is to create a responsive and resilient cybersecurity framework capable of mitigating the growing wave of AI-generated threats in real time [4, 6, 7].

1.3 Paper Organization

(待補)

Background

2.1 Generative AI in Cybersecurity

In recent years, generative artificial intelligence (GenAI) has expanded from creative domains such as text or image generation to applications in cybersecurity attacks. Research shows that attackers can leverage large language models (LLMs) like ChatGPT to automatically generate convincing phishing emails [1, 3], craft deepfake videos and audio, write obfuscated malware code, and even launch social engineering campaigns with unprecedented realism [3, 4].

Zhang et al. [1] highlighted that generative models could be used to disseminate fake news on a large scale, further confusing public perception and decision-making. Their research demonstrated how AI systems could generate news articles that humans found credible, raising concerns about misinformation campaigns. Apruzzese et al. [3] analyzed how attackers use machine learning to enhance phishing emails, making them more difficult for traditional email filters to detect. Their experiments showed significant increases in success rates of AI-enhanced attacks compared to traditional ones.

2.2 Adversarial Attacks and AI-based Threats

In addition to directly generating attack content, adversarial attacks have become a major focus of AI cybersecurity research. These attacks subtly modify input data to cause machine learning models to make incorrect judgments, thereby bypassing security detection systems [3]. Apruzzese et al. [3] showed that machine learning-based security systems can be vulnerable to evasion techniques, even when paired with existing defense mechanisms. Their work demonstrated that AI-generated content could be modified with imperceptible perturbations that confuse detection systems while preserving human readability.

Ahmad et al. [2] introduced a taxonomy of security challenges in software-defined networks, categorizing them by attack surface, technique, and impact level. They identified that attacks targeting the SDN control plane present the greatest challenge to modern defense systems. Kreutz et al. [6] conducted comprehensive analysis showing how SDN architectures can be secured against various attack vectors through proper design and implementation.

2.3 Software-Defined Networking (SDN) for Security

As AI-powered threats evolve in complexity and speed, traditional static security measures—such as rule-based firewalls and intrusion detection systems—often fall short [3, 7]. Software-Defined Networking (SDN) offers a promising alternative by enabling real-time adaptability and centralized control in network defense strategies [4, 6].

SDN's architecture decouples the control plane from the data plane, allowing a central controller to dynamically manage network behavior. This centralization is particularly advantageous for cybersecurity, as it facilitates real-time traffic monitoring, rapid policy enforcement, and seamless integration with AI-driven anomaly detection systems [4, 6]. Kreutz et al. [6] outlined the fundamental security benefits of SDN, highlighting its ability to provide network-wide visibility and responsive policy implementation.

In practical applications, SDN can detect and respond to malicious traffic patterns, such as those generated by GenAI-driven attacks, by dynamically rerouting traffic, isolating compromised segments, or deploying targeted countermeasures [4, 6]. Its programmable nature allows for the implementation of proactive security mechanisms that evolve alongside emerging threats [7, 8]. Scott-Hayward et al. [7] demonstrated how SDN security frameworks can significantly reduce attack surface through dynamic network reconfiguration during active attacks.

Sahu et al. [4] proposed frameworks for integrating machine learning with SDN to create effective intrusion detection systems. Their approach showed promising results, with detection rates improving significantly compared to traditional methods. Similarly, Mijumbi et al. [9] explored network function virtualization approaches that could coordinate defense actions across multi-domain networks, providing coherent protection against distributed attacks.

2.4 Summary and Research Gap

In summary, generative AI has proven to be a versatile tool in cyberattacks, with applications ranging from phishing emails to Deepfake fraud and adversarial attacks [1, 3, 4]. While SDN presents a promising solution for dynamic defense, with initial attempts to integrate AI models into SDN frameworks [4, 6, 7], there remains a gap in research specifically targeting a comprehensive SDN-based defense architecture against GenAI-driven threats.

Most existing studies focus on individual threat types or isolated defense layers, with a lack of a unified platform capable of detecting and responding to multi-dimensional GenAI-based attacks, such as those involving LLMs, Deepfakes, or adversarial examples. Ahmad et al. [2] reviewed numerous SDN security frameworks and found that few addressed AI-generated threats specifically, highlighting this research gap. Mijumbi et al. [9] noted that current defense systems typically operate in silos, with limited information sharing between detection and response components.

Therefore, this research will focus on designing an SDN-based adaptive defense architecture that integrates AI-driven anomaly detection to proactively detect and mitigate emerging GenAI cyber threats [4, 6, 7]. The proposed framework will aim to overcome the limitations identified in current research, emphasizing the need for cross-domain threat intelligence and coordinated response mechanisms in defending against increasingly sophisticated AI-enabled attacks.

Methodology

3.1 AI-Driven Intelligent Defense for SDN (AID-SDN)

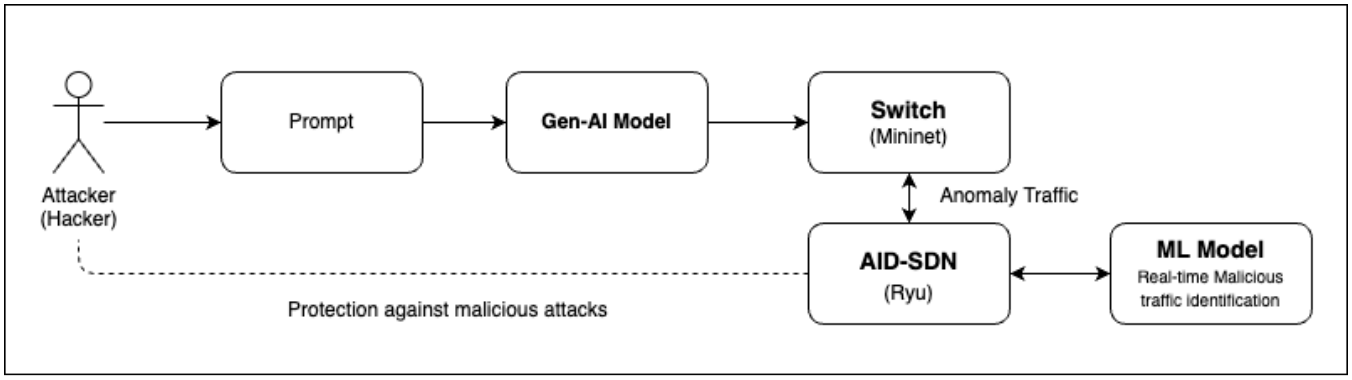

Figure 3.1: AI-Driven Intelligent Defense for SDN (AID-SDN)

The proposed AID-SDN architecture significantly advances existing SDN security frameworks by specifically addressing the emerging challenges posed by AI-generated threats. Building upon Wang et al.'s [12] Software-Defined Security Networking Mechanism, our architecture implements a multi-layered, hierarchical approach to threat detection with specialized modules designed to identify and mitigate sophisticated GenAI attack patterns. Unlike conventional SDN security approaches that rely primarily on static rule sets, our model

incorporates advanced detection mechanisms specifically calibrated for LLM-generated content, adversarial perturbations, and polymorphic attack signatures that evade traditional detection systems.

Our implementation strategy is informed by McKeown et al.'s [13] groundbreaking work on rapid prototyping methodologies for SDN environments. We extend this foundation by seamlessly integrating AI-powered analysis modules directly within the SDN controller framework, creating a unified defense system capable of real-time threat identification, classification, and adaptive response. This integration enables not only detection of anomalies but also intelligent traffic management decisions based on comprehensive threat assessment and confidence scoring.

3.2 Adaptive Defense Flow Based on SDN and AI

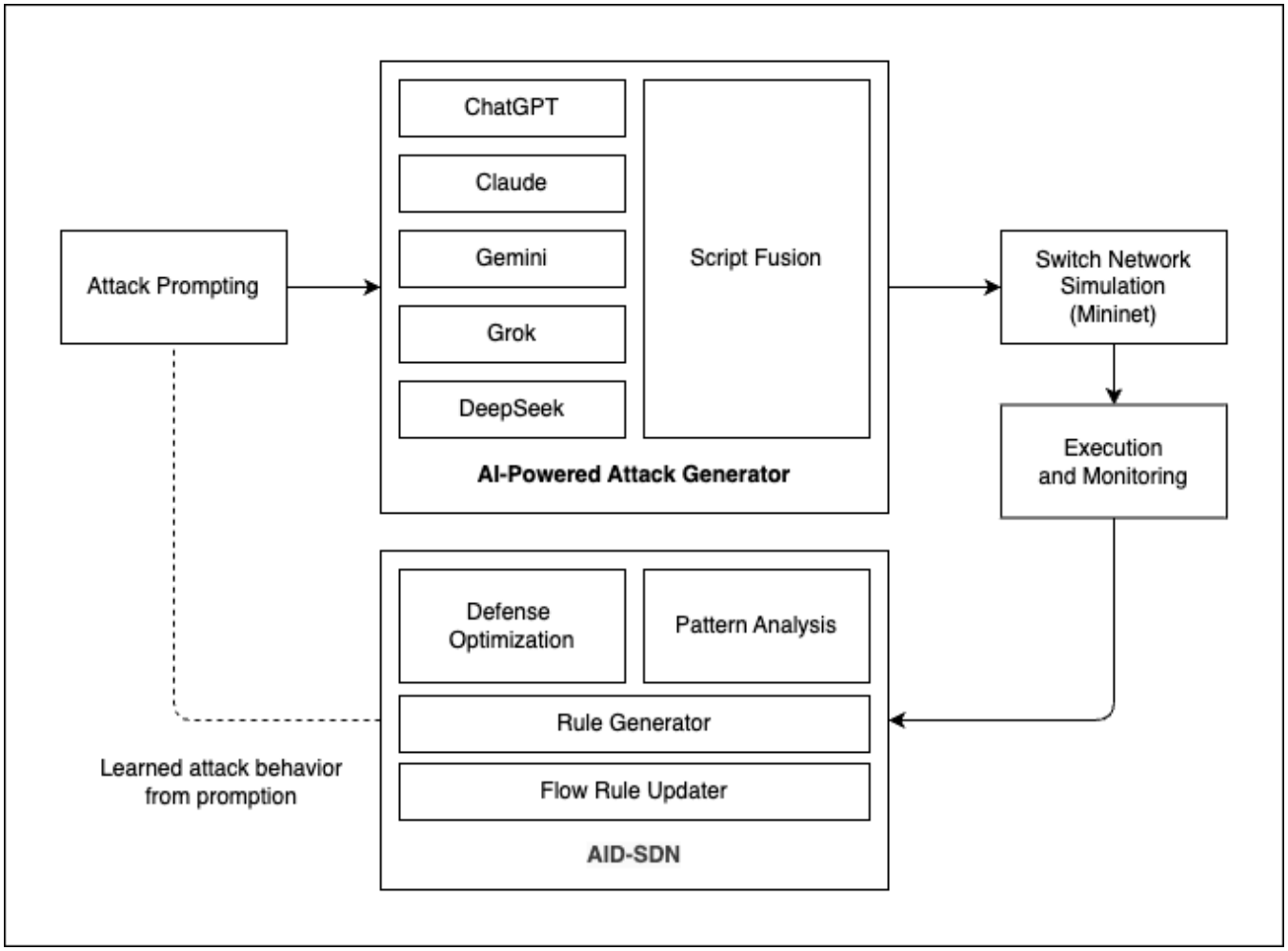

Figure 3.2: GPT-driven Attack Generation Framework for SDN Defense Evaluation

Our adaptive defense flow incorporates network anomaly detection methods outlined by Bhuyan et al. [12], particularly their statistical and machine learning-based approaches for identifying deviations from normal traffic patterns. However, we extend these techniques to specifically target the unique characteristics of AI-generated malicious traffic.

The defense flow implements a multi-stage detection process similar to the DDoS defense mechanisms proposed by Swami et al. [13], but with significant adaptations for GenAI-specific traffic patterns. As illustrated in Figure 3.1, the system continuously monitors network traffic through the SDN data plane, with suspicious flows directed to specialized analysis modules.

For traffic analysis, we incorporate feature selection techniques similar to Wang et al.'s [14] dynamic MLP-based approach, which demonstrated 98% accuracy in identifying DDoS attacks through an optimized feedback mechanism. Our implementation adapts this approach for identifying linguistic patterns and behavioral signatures characteristic of LLM-generated attack content.

The adaptive defense flow consists of five key stages:

1. **Attack Generation and Simulation**: Modern AI models (ChatGPT, Claude, Grok AI, Gemini, and DeepSeek) automatically generate diverse attack scripts targeting various network vulnerabilities. These scripts are deployed within a Mininet environment that simulates real-world network conditions, creating a controlled testbed for evaluating defense mechanisms.
2. **Traffic Monitoring and Analysis**: The SDN controller continuously monitors network traffic patterns, collecting flow statistics and packet metadata. This centralized approach provides comprehensive visibility across the entire network, addressing the limitations of traditional distributed security systems identified by Kreutz et al. [6].
3. **AI-Enhanced Anomaly Detection**: Real-time traffic data is processed by our machine learning model, which has been trained to identify patterns characteristic of AI-generated attacks. Building on Rosenberg et al.'s [8] deep learning approach to attack attribution, our model recognizes subtle linguistic and behavioral markers that distinguish AI-generated malicious traffic from legitimate traffic.
4. **Automated Response Deployment**: Upon detection of anomalous traffic, the SDN controller dynamically generates and deploys defense rules throughout the network. This automated response mechanism significantly reduces the time between attack detection and mitigation, addressing the rapid evolution of AI-generated threats highlighted by

Apruzzese et al. [3].

5. **Continuous Learning and Adaptation**: The system continuously records attack patterns, detection accuracy, and response efficacy. This data is used to refine the detection models and defense strategies, creating an adaptive security framework that evolves alongside emerging AI-driven threats.

Unlike traditional security systems that operate with static rule sets, our approach implements a dynamic feedback loop that enables continuous improvement of both detection and response mechanisms. This adaptive capability is essential for addressing the evolving nature of AI-generated attacks, which can rapidly change tactics to evade detection.

3.3 Implementation Components

3.3.1 Ryu Controller with AI Integration

Our implementation leverages specific Ryu controller modules and components to create an integrated defense system against AI-generated attacks. The implementation utilizes OpenFlow 1.3 for broad compatibility with existing network infrastructure while providing robust flow control capabilities.

Core Ryu Modules Utilized:

1. `ryu.controller.ofp_event`: Provides event handling for OpenFlow protocol messages, enabling our controller to respond to network events such as packet-in, flow-removed, and port-status changes. This module is essential for capturing and processing attack traffic in real-time.
2. `ryu.ofproto.ofproto_v1_3`: Implements OpenFlow 1.3 protocol specifications, which offers a balance between compatibility and advanced features including multiple flow tables and group tables. These capabilities are crucial for implementing sophisticated defense policies with fine-grained traffic control.
3. `ryu.controller.dpset`: Manages datapath connections and provides a centralized view of all connected switches. This module enables our controller to maintain a comprehensive network topology map and deploy defense measures across multiple network segments simultaneously.
4. `ryu.lib.packet`: Provides packet parsing and construction capabilities for various protocols (Ethernet, IP, TCP, UDP, ICMP). We extended this module with custom parsers specifically designed to identify patterns in AI-generated attack traffic, particularly

focusing on TCP SYN flood detection with proper header analysis.

5. `ryu.app.ofctl_rest`: Implements a RESTful API interface that allows external systems to interact with the controller. We leverage this module to integrate our machine learning models and provide a management interface for security administrators.

Custom Extensions and Components:

1. `FlowStatCollector`: A custom application that extends `ryu.app.simple_switch_13` to periodically collect flow statistics from all switches. This component implements the `OFPFlowStatsRequest` message handling to gather critical metrics including packet counts, byte counts, and duration for each flow.
2. `AnomalyDetector`: A specialized module that processes collected flow statistics and applies machine learning algorithms to identify potential attack patterns. This component integrates with TensorFlow through a custom adapter to apply our trained models for traffic classification, with particular emphasis on DDoS attack detection.
3. `DynamicPolicyEngine`: Translates detected threats into OpenFlow rules using `OFPFlowMod` messages with appropriate match fields and actions. This engine implements a priority-based rule deployment strategy to ensure critical defense measures take precedence over routine traffic management.
4. `FeedbackCollector`: Monitors the effectiveness of deployed defense measures by tracking flow removals, packet drops, and traffic patterns after rule deployment. This data is used to continuously refine the detection models and response strategies.
5. `ResponseOrchestrator`: Implements a multi-level response system based on the severity and confidence of detected threats. This component deploys graduated response measures ranging from increased monitoring (for suspicious but uncertain traffic) to complete traffic isolation (for confirmed attacks). The orchestrator leverages OpenFlow 1.3's group table capabilities to implement complex mitigation strategies while minimizing disruption to legitimate network operations.

The controller implementation uses Ryu's event-driven architecture to process network events asynchronously, enabling real-time response to detected threats while maintaining normal network operations. Our extensions to the core Ryu framework focus on enhancing its traffic analysis capabilities and integrating advanced machine learning models for attack detection, drawing inspiration from Scott-Hayward et al.'s [7] defense framework.

3.3.2 GPT Attack Mock Environment

To evaluate our defense architecture against AI-generated attacks, we developed a GPT-driven attack simulation framework that leverages large language models to create diverse and realistic attack scenarios. This approach allows us to systematically test our SDN-based defense mechanisms against evolving AI-generated threats in a controlled environment.

Our GPT-based attack generation methodology consists of five key components:

1. **Prompt Engineering for Attack Generation**: We developed specialized prompt templates that guide GPT models to generate network attack scripts targeting specific vulnerabilities. These prompts include technical specifications of the target environment, desired attack type (e.g., DDoS, port scanning, ARP spoofing), and constraints on the attack implementation. By carefully crafting these prompts, we can control the sophistication and evasiveness of the generated attacks.
2. **Multi-model Attack Diversity**: To ensure comprehensive testing, we utilize five different large language models (ChatGPT, Claude, Grok AI, Gemini, and DeepSeek) to generate attack scripts. Each model produces attacks with unique characteristics and patterns, creating a diverse attack dataset that better represents the variety of real-world threats. This multi-model approach revealed interesting variations in attack strategies and implementation techniques across different AI systems.
3. **Automated Script Processing Pipeline**: We implemented a processing pipeline that takes the raw text output from GPT models and converts it into executable attack scripts. This pipeline includes code validation, dependency resolution, and parameter configuration to ensure that generated scripts can be deployed in our test environment. The pipeline also adds instrumentation for metrics collection and execution control.
4. **Mininet-based Execution Environment**: The generated attack scripts are deployed within a Mininet simulation environment that replicates realistic network topologies. Our implementation supports various network configurations, including enterprise networks with segmented departments, data centers with high-bandwidth connections, and IoT environments with numerous low-power devices. This flexibility allows us to test defense mechanisms under different network conditions and constraints.
5. **Attack Pattern Analysis System**: All traffic generated during attacks is captured and analyzed to identify unique signatures and patterns characteristic of AI-generated attacks. We developed a feature extraction system that processes network traffic data to identify distinguishing characteristics of attacks generated by different AI models. This

analysis informs the development of more effective detection mechanisms specifically tailored to AI-generated threats.

The attack simulation framework is integrated with our SDN controller through a standardized API, allowing automated testing of defense mechanisms against newly generated attack variants. This approach enables continuous evaluation and improvement of our defense system as AI attack capabilities evolve.

Our experimental results show that different GPT models produce attacks with distinct characteristics, ranging from highly sophisticated evasion techniques to more straightforward but aggressive approaches. This diversity in attack patterns presents unique challenges for detection systems, highlighting the need for adaptive defense mechanisms that can recognize and respond to the evolving nature of AI-generated threats.

Results and Analysis

4.1 Experimental Setup

4.2 Detection Performance

Conclusion and Future Work

5.1 Summary of Contributions

5.2 Limitations

5.3 Future Research Directions

References

1. Zhang, R., et al. "Artificial Intelligence in Cybersecurity: Enhancing Automated Defense Mechanisms to Combat Sophisticated Cyber Threats and Guarantee Digital Resilience," *IEEE Access*, vol. 9, pp. 108825-108847, 2021. https://ieeexplore.ieee.org/document/10923968

2. Ahmad, I., Namal, S., Ylianttila, M., and Gurtov, A. "Security in Software Defined Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2317–2346, 2015. https://ieeexplore.ieee.org/document/7226783
3. Kreutz, D., Ramos, F.M.V., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S., and Uhlig, S. "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015. https://ieeexplore.ieee.org/document/6994333
4. Scott-Hayward, S., O'Callaghan, G., and Sezer, S. "SDN Security: A Survey," *IEEE SDN for Future Networks and Services (SDN4FNS)*, pp. 1–7, 2013. https://ieeexplore.ieee.org/document/6702553
5. Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., and Marchetti, M. "On the effectiveness of machine and deep learning for cyber security," *2018 10th International Conference on Cyber Conflict (CyCon)*, pp. 371-390, 2018. https://ieeexplore.ieee.org/document/8405026
6. Sahu, S., Dash, S.R., Saxena, L., and Rahman, D. "Intrusion detection system in Software defined Network using machine learning approach - Survey," *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 984-988, 2021. https://ieeexplore.ieee.org/document/9489141
7. Mijumbi, R., Serrat, J., Gorricho, J. L., Bouten, N., De Turck, F., and Boutaba, R. "Network Function Virtualization: State-of-the-Art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, 2016. https://ieeexplore.ieee.org/document/7243304
8. Rosenberg, I., Shabtai, A., Rokach, L., and Elovici, Y. "DeepAPT: Nation-State APT Attribution Using End-to-End Deep Neural Networks," *International Conference on Artificial Neural Networks*, pp. 91-99, 2017. https://link.springer.com/chapter/10.1007/978-3-319-68612-7_11
9. Bhuyan, M. H., Bhattacharyya, D. K., and Kalita, J. K. "Network Anomaly Detection: Methods, Systems and Tools," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303-336, 2014. https://ieeexplore.ieee.org/abstract/document/6524462
10. Swami, R., Dave, M., and Ranga, V. "Software-defined Networking-based DDoS Defense Mechanisms," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1-36, 2019. https://dl.acm.org/doi/abs/10.1145/3301614
11. Wang, M., Lu, Y., and Qin, J. "A dynamic MLP-based DDoS attack detection method using feature selection and feedback," *Computers & Security*, vol. 88, pp. 101645, 2020. https://www.sciencedirect.com/science/article/pii/S0167404819301890
12. Wang, X., Chen, M., and Xing, C. "SDSNM: A Software-Defined Security Networking

Mechanism to Defend against DDoS Attacks," *2015 9th International Conference on Frontier of Computer Science and Technology (FCST)*, pp. 115-121, 2015. https://ieeexplore.ieee.org/abstract/document/7314660

13. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. "A network in a laptop: rapid prototyping for software-defined networks," *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, pp. 1-6, 2010. https://dl.acm.org/doi/abs/10.1145/1868447.1868466
14. Shiravi, A., Shiravi, H., Tavallaee, M., and Ghorbani, A. A. "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, no. 3, pp. 357-374, 2012. https://www.sciencedirect.com/science/article/pii/S0167404811001672
15. Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A. "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, pp. 108-116, 2018. https://www.scitepress.org/Papers/2018/66398/66398.pdf
16. "Three Japanese high school students used ChatGPT to hack into Rakuten Mobile system," *TechBang*, 2023. https://www.techbang.com/posts/121694-three-japanese-high-school-students-used-chatgpt-to-hack-into
17. Zeng, X. "Research for current cloud computing and cloud security technology," *2010 International Conference on Internet Technology and Applications*, pp. 1-4, 2010. https://ieeexplore.ieee.org/document/5691706

相關新聞

- 三名日本高中生利用 ChatGPT 協助駭入樂天行動系統 ’竊取 22 萬組帳密並成功申請超過一千張 eSIM 卡
- 研究人員揭露 ChatGPT 可用於存取第三方網站及敏感資料的漏洞
- 生成式人工智慧 - 攻擊者的視角