1. Introduction: Artificial intelligence is being widely used in life, which brings greater risks of cyber attack and leads to problems like personal data leakage which concerns all of us. We want to understand what offences against generative AI and specifically large-language models exist, how they work, and what are the mitigations. We therefore set out to conduct a survey on this topic.
2. Background: Artificial Intelligence (AI) refers to computer learning models that can perform human intelligence tasks at a much faster rate, including: learning, reasoning and problem solving. Our research focuses on 2 types of AI: Large Language Models (LLMs) and Generative AI(GenAI). LLMs refer to machine learning models that are trained on textual data. LLMs, the most popular being ChatGPT, take in textual data and give a text response. GenAI is a broader definition of AI that includes synthesizing and generating other media outside of but also including text, such as code or images. The key distinction between the two is that LLMs focus on natural language processing, GenAI has a wider range of generative capabilities, and that all LLMs are GenAI, but not all GenAI are LLMs.
3. Challenges of Generative AIs
    - 3.1. Offense
        - 3.1.1. Jailbreaking: Wei (2023) argued that Generative AIs are vulnerable to jailbreaking because of two key failure modes of safety training: competing objectives and mismatched generalisation, and argued that generative AIs' safety training should be on par with their inner mechanism.
        - 3.1.2. Membership Inference Attack: Duan et al (2023) demonstrated that existing MIA methods are ineffectiveness to diffusion model based generative AI, proposed the Step-wise Error Comparing Membership Inference (SecMI) that is a query-based MIA that can infer memberships of the data in the training data set by assessing the matching of forward process posterior estimation at each timestep, and showed that diffusion model is weak this SecMI and suffered from great privacy risk. (Xingyu)
        - 3.1.3. Gen AI enables phishing, social engineering, and malware creation - Gupta (2023)
        - 3.1.4. Attackers exploit jailbreaking, prompt injections, and role-play manipulation to bypass restrictions - Gupta (2023)
        - 3.1.5. AI deep fakes and voice cloning: GenAI can be used to generate non-consensual images and voices of people or other objects - Marchal et al (2024)
        - 3.1.6. Impersonation: GenAI can take in data and accurately replicate personalities of a person through text, which can lead to phishing and scams - Marchal et al (2024)

3.1.7. Illegal Monetization: GenAI can create non-consensual intimate imagery of both adults and children which can then be sold, which is both highly illegal and unethical - Marchal et al (2024)

3.1.8. Disinformation: GenAI has been used to create political propaganda and spread disinformation about others - Marchal et al (2024)

3.2. Privacy / Data Leakage

3.2.1. Generative AI can leak sensitive data through poorly constrained responses - Gupta (2023)

3.2.2. Ethical concerns in AI-generates misinformation and fraud - Gupta (2023)

3.2.3. Potential for adversarial attacks to extract training data - Gupta (2023)
Prompt Injection can be used to reveal information not intended for the user, such as instruction given to the chat bot - Gupta (2023)
Figure of Example is in Paper

3.2.4. Model inversion attacks allow attackers to extract training data from a model. Additionally, AI can generate summaries of private conversations, which leads to security concerns for data exposure - Marchal et al (2024)

4. Challenges of LLMs

4.1. Offense

4.1.1. Dataset poisoning: Carlini(2024) demonstrates how it is highly practical for attackers to manipulate real world datasets to poison the model.

4.1.2. Code generation: LLM-generated code may introduce security vulnerabilities (ex CWE 787: out-of-bounds writes) - Sandoval(2023)
LLMs may be trained over insecure/buggy code and reproduces these errors in it's own code. Code generated by LLMs may be secure independently but insecure when working with other code - Sandoval (2023)
40% of Code Suggestions by Github CoPilot contain security related bugs - Sandoval (2023)

4.1.3.

Remote code execution (RCE): Liu et al (2024) developed LLMSmith to detect and exploit RCE vulnerabilities in LLM integrated apps and frameworks. RCE vulnerabilities allow attackers to execute arbitrary code or even obtain full access to the apps through prompt injection alone. LLMSmith found 20 vulnerabilities in 11 frameworks and the researchers successfully attacked 17 out of 51 apps with their prompt-injection based method.
LLMs can obfuscate malicious code, which makes detection harder. Additionally, malware generated by LLMs bypassed 9 out of 12 antivirus softwares-Motlag et al (2024)

4.1.4. Attackers exploit LLMs to generate insecure code snippets - Pearce(2023)

4.1.5. Backdoor threats in LLM-based agents (ex triggering unintended behaviors) - Yang (2024)
Backdoor attacks on LLM-based agents can manipulate intermediate reasoning steps, making them harder to detect compared to traditional LLM backdoor attacks. - Yang (2024)

4.1.6. Remote code execution (RCE): Liu et al (2024) developed LLMSmith to detect and exploit RCE vulnerabilities in LLM integrated apps and frameworks. RCE vulnerabilities allow attackers to execute arbitrary code or even obtain full access to the apps through prompt injection alone. LLMSmith found 20 vulnerabilities in 11 frameworks and the researchers successfully attacked 17 out of 51 apps with their prompt-injection based method.

4.1.7. GPT-4 outperformed humans in Capture the Flag (CTF) challenges, and achieved 78% accuracy with a human-in-the-loop design. Overall, LLMs completed cryptography challenges at an 85% success rate, and web exploitation challenges at an 80% success rate, outperforming humans. - Shao et al (2024)

4.1.8. PassGPT, an LLM trained on leaked passwords, could significantly outperform brute-force attacks when attempting to access a password. Additionally, WorkGPT and FraudGPT are models trained to automate scam and phishing emails based on existing data. - Motlag et al (2024)

4.1.9. LLMs can obfuscate malicious code, which makes detection harder. Additionally, malware generated by LLMs bypassed 9 out of 12 antivirus softwares-Motlag et al (2024)

4.2. Privacy / Data Leakage

4.2.1. LLMs trained on public repositories may expose sensitive data - Pearce(2023)

4.2.2. Potential leakage of proprietary or confidential information in responses - Pearce(2023)

4.2.3. Adversaries can extract information through prompt injection attacks - Gupta (2023)

4.2.4. LLMs can expose Personally Identifiable Information if they are not configured correctly. However, when configured correctly, OneShield Privacy Guard flagged 8.25% of 1256 GitHub pull requests that contained PIIs. - Asthana et al(2025)

4.2.5. Prompt injection may allow attackers to extract confidential information, and LLM responses may accidentally reveal sensitive information

4.3. Defense

4.3.1. Defense against RCE: Liu et al (2024) suggested that developers should set users' access level to the lowest to avoid attackers gaining access to the apps, isolate the code generated by LLM to avoid execution of untrusted code, and possibly detect malicious prompts.

4.3.2. LLMS can assist in secure code generation and vulnerability detection - Pearce (2023)

4.3.3. Use in cybersecurity automation (ex automatic patching & incident response) - Gupta (2023)
ChatGPT can generate cybersecurity incidents, threat intelligence, vulnerability assessments based on cybersecurity data from social media, news articles, dark web forums, etc. - Gupta (2023)
When implemented in intrusion detection systems, ChatGPT can identify threats and generate descriptions of the attack behavior by processing network logs and SIEM alerts. - Gupta (2023)
ChatGPT can relieve analyst's workload by automatically analyzing logs for abnormal behavior and providing recommendations for next steps. - Gupta (2023)

4.3.4. Need for better filtering and security aware training - Gupta (2023)

4.3.5. Traditional Personally Identifiable Information (PII) detection tools such as StarPII and Presidio struggle in context-dependent or multilingual situations since they are static models. OneShield Privacy Guard is an ML-based tool which got a 95% F1 score across 26 different languages and saved 300+ hours of manual review time in testing. - Asthana et al(2025)

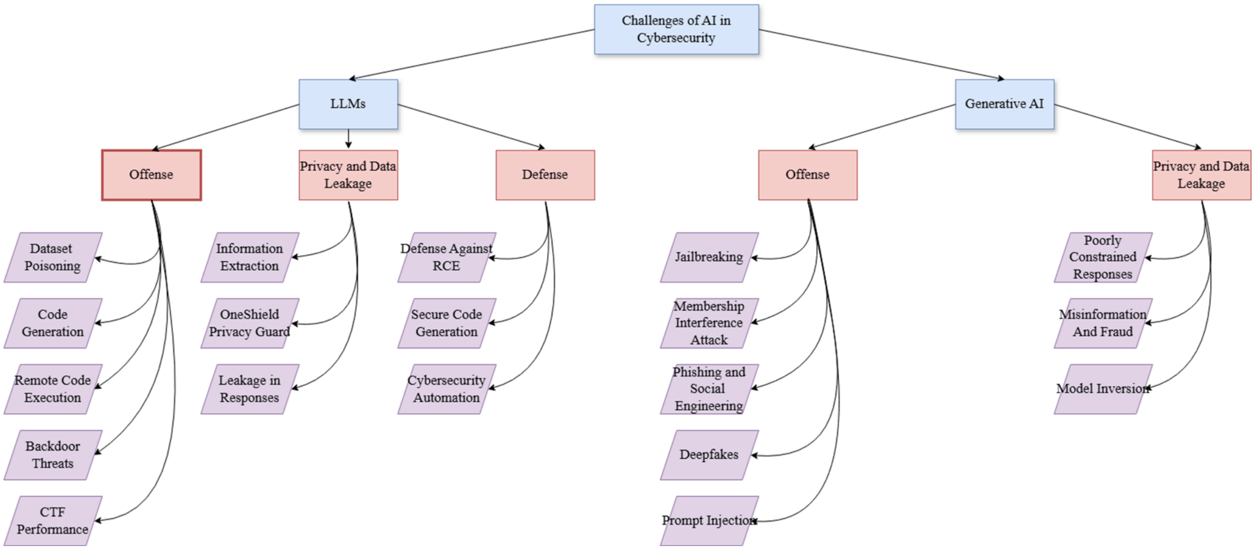
4.3.6. LLMs process security logs at a faster and more accurate rate than humans. They also reduce false positives, enhance SIEM security monitoring and can train cybersecurity teams by generating CTF questions. -Motlag et al (2024)

4.3.7. LLMs struggle to recognize novel threats, and generated security threats can sometimes be incorrect or misleading. -Motlag et al (2024)

4.3.8. Generative AI models can be attacked through: Model extraction by repeatedly querying APIs, data poisoning by injecting malicious data, and prompt injection to bypass safety filters of an AI model - Marchal et al (2024)

5. Figures:

5.1 Figure 1: Resulting Offense and Defense Vectors



5.2 Figure 2: Table of Paper Coverage

Papers	**GenAI**	**LLM**	**Offences**	**Defences**	**Privacy/Data Leakage**
Asthana et al (2025)		✓		One Shield Privacy Guard	Peronsal Identifiable Information Leakage
Calini et al (2024)	✓		Data Poisoning	SHA-256 Hash, Vary Snapshot Time	
Duan et al (2023)	✓		Membership Inference Attack	Data Augmentation	Peronsal Identifiable Information Leakage
Gupta et al. (2023)	✓		Attack Vector Creation	Automated Incident Response,	Prompt Injection
Liu et al (2024)		✓	Remote Code Execution	Permission Management, Environment Isolation, Prompt Analysis	
Marchal (2024)	✓		Deepfakes, Prompt Injection, Data Poisoning		Model Extraction, Private Conversations
Motlag (2024)		✓	PassGPT, Malware Generation	Process Security Logs	
Pearce et al. (2023)		✓			Information Leakage
Sandoval et al. (2023)		✓	Generation of Insecure Code		
Shao et al (2024)		✓	CTF Performance	Train Security Teams, Vulnerability detection	
Wei (2023)		✓	Jailbreaking	Safety-Capability Parity*	
Yang et al. (2024)		✓	Backdoor Attacks		

6. Conclusion: The integration of AI into our daily life comes with significant cyber security challenges. Forms of attack include jailbreaking, data poisoning, remote code execution, backdoor attacks, and so on, which brings great risk of privacy violation. While AI enhances many aspects of our life, it also broadens the attack surface. It is vital to understand how to respond to these threats. We have learned that they can be addressed through stronger safety training, improved model transparency, and robust defense mechanisms. Future research should focus on secure AI model development, adversarial defense techniques, and regulatory frameworks to ensure we deplore AI in a responsible and secure way.

7. References

[1] S. Ullah, M. Han, S. Pujar, H. Pearce, A. Coskun, and G. Stringhini, “Can Large

Language Models Identify And Reason About Security Vulnerabilities? Not Yet." Available: https://www.bu.edu/peaclab/files/2024/01/saad_arxiv.pdf

[2] W. Yang, X. Bi, Y. Lin, S. Chen, J. Zhou, and X. Sun, "Watch Out for Your Agents! Investigating Backdoor Threats to LLM-Based Agents," *arXiv.org*, Oct. 29, 2024. https://arxiv.org/abs/2402.11208

[3] H. Pearce, B. Tan, B. Ahmad, R. Karri, and B. Dolan-Gavitt, "Examining Zero-Shot Vulnerability Repair with Large Language Models," *2023 IEEE Symposium on Security and Privacy*, May 2023, doi: https://doi.org/10.1109/sp46215.2023.10179324.

[4] M. Gupta, C. Akiri, K. Aryal, E. Parker, and L. Praharaj, "From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy," *IEEE Access*, vol. 11, pp. 80218–80245, Aug. 2023, doi: https://doi.org/10.1109/ACCESS.2023.3300381.

[5] G. Sandoval, H. Pearce, T. Nys, R. Karri, S. Garg, and B. Dolan-Gavitt, "Lost at C: A User Study on the Security Implications of Large Language Model Code Assistants," *www.usenix.org*, 2023. https://www.usenix.org/conference/usenixsecurity23/presentation/sandoval

[6] J. Duan, F. Kong, S. Wang, X. Shi, and K. Xu, "Are Diffusion Models Vulnerable to Membership Inference Attacks?," in *Proceedings of the 40th International Conference on Machine Learning*, PMLR, Jul. 2023, pp. 8717–8730. Accessed: Jan. 31, 2025. [Online]. Available: https://proceedings.mlr.press/v202/duan23b.html

[7] N. Carlini *et al.*, "Poisoning Web-Scale Training Datasets is Practical," May 06, 2024, *arXiv*: arXiv:2302.10149. doi: 10.48550/arXiv.2302.10149.

[8] T. Liu, Z. Deng, G. Meng, Y. Li, and K. Chen, "Demystifying RCE Vulnerabilities in LLM-Integrated Apps," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, in CCS '24. New York, NY, USA: Association for Computing Machinery, Dec. 2024, pp. 1716–1730. doi: 10.1145/3658644.3690338.

[9] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How Does LLM Safety Training Fail?," Jul. 05, 2023, *arXiv*: arXiv:2307.02483. doi: 10.48550/arXiv.2307.02483.

[10] S. Asthana, B. Zhang, R. Mahindru, C. DeLuca, A. L. Gentile, and S. Gopisetty, "Deploying Privacy Guardrails for LLMs: A Comparative Analysis of Real-World Applications," *arXiv.org*, 2025. https://arxiv.org/abs/2501.12456

[11] N. Marchal, R. Xu, R. Elasmar, I. Gabriel, B. Goldberg, and W. Isaac, "Generative AI Misuse: A Taxonomy of Tactics and Insights from Real-World Data," *arXiv.org*, Jun. 21, 2024. https://arxiv.org/abs/2406.13843

[12] M. Shao *et al.*, "An Empirical Evaluation of LLMs for Solving Offensive Security Challenges," *arXiv.org*, 2024. https://arxiv.org/abs/2402.11814

[13] F. N. Motlagh, M. Hajizadeh, M. Majd, P. Najafi, F. Cheng, and C. Meinel, "Large Language Models in Cybersecurity: State-of-the-Art," *arXiv.org*, Jan. 30, 2024. https://arxiv.org/abs/2402.00891