XXI Escola Regional de Redes de Computadores / VIII Workshop de Segurança (ERRC). DOI: 10.5753/errc.2024.4669

Artigo Completo/Full Paper

Estudo comparativo do uso de LLM para atividades de Red Team em LLM

Comparative study of the use of LLM for Red Team activities in LLM

Ana C.V. Alves • ana.cva@edu.udesc.br
Universidade do Estado de Santa Catarina (UDESC)
Charles C. Miers • charles.miers@udesc.br
Universidade do Estado de Santa Catarina (UDESC)

resumo. O uso de *Large Language Model*s (LLMs) cada vez mais intensivo e com expressivos parâmetros e entendimento de contexto pode ser explorado para causar problemas de segurança. Este trabalho visa esclarecer uma comparação entre as ferramentas disponíveis para detecção de vulnerabilidades no escopo de *Red Teaming* em LLMs. Foi realizada uma revisão sistemática, bem como comparações de algoritmos de estado da arte na área com novas propostas o uso de LLMs de forma adversária a algoritmos da mesma forma, uma nova subárea de pesquisa de *jailbreaks* é revisada neste artigo.

abstract. The rising of the use of LLMs with models with increasingly more parameters and context understanding, this work aims to clarify a comparison between the tools available for vulnerability detection in the scope of *Red Teaming* in LLMs. A systematic review of the topic was carried out and with comparisons of state-of-the-art algorithms in the area with new proposals for the use of LLMs in an adversarial way to algorithms in the same way, a new research sub-area of *jailbreaks* is reviewed in this article.

palavras-chave: IA Generativa • Red Team • LLM • Segurança
keywords: Generative AI • Red Team • LLM • Security

1 Introdução

A área de *Generative Artificial Intelligence* (GenAI) teve uma nova classificação sobre a capacidade de geração de texto em línguas naturais através de modelos de larga escala chamados *Large Language Model*s (LLMs). Isso foi possível devido às tecnologias como a arquitetura proposta em modelos como o *Generative Pre-trained Transformer* (GPT), com um poder de processamento e entendimento de contexto superior aos antigos modelos como ELMo [1].

A nova escala de LLMs é classificada pela quantidade de parâmetros e hiper-parâmetros que estes algoritmos conseguem processar [2]. Cada rede neural é composta por camadas de neurônios, que dada suas funções próprias, respondem valores a cada troca de informações na arquitetura, definidos pelos dados de entrada. Estas informações são resultados de outros atributos pré-estruturados pela etapa de treinamentos da rede, chamados como hiper-parâmetros [3].

O LLM é empregado em várias aplicações na Internet atualmente, sendo que as instituições bancárias têm sofrido pressão para essa transformação digital com integração de serviços bancários com Inteligência Artificial (IA) Generativa do tipo LLM [4]. Diversos bancos brasileiros, como o Banco do Brasil desde de o ano de 2022, desenvolvem soluções com IA [5], e o Banco Bradesco implementa soluções de GenAI por *prompts* textuais com LLMs com a sua assistente virtual BIA [6].

Em um cenário em que aplicações na Internet utilizam essas ferramentas, existe a incerteza sobre as garantias de respostas não maliciosas destas, dada a natureza desse tipo de algoritmo com resultados não claramente previsíveis, resultando em possíveis "papagaios estocásticos" [7]. A capacidade de filtrar as informações desde a fonte de dados de treinamento, até possíveis respostas incorretas é um desafio atual, pois LLMs treinados com *datasets* abrangendo conteúdos prejudicais são vulneráveis, assim como aqueles sujeitos a *jailbreaks* [8]. O vazamento de informações usadas no treinamento, com modelos que utilizam a Internet como fonte, podem apresentar um risco de exposição de dados sensíveis, além de vieses não controlados, como *jailbreaks* de 2022 mostram em modelos como GPT-3 [9]. Com intuito de identificar os riscos de LLMs, diversas estratégias para mitigação de erros são propostas [10], com exames extensivos sobre a importância de ataques adversários para melhorar os modelos de processamento de linguagem natural, com testes de

Todos os autores são membros da SBC.

ataques em um ambiente controlado. Dentro do contexto de modelos adversariais, há ferramentas que utilizam esse tipo de estratégia com LLMs para detecção de vulnerabilidades, bem como avaliação destas, além de auxiliar na análise de riscos. Utilizando princípios de *Red Teaming* os modelos ajustam *prompts* maliciosos através de aprendizado de máquina até terem uma taxa de acerto satisfatória.

O objetivo deste artigo é comparar *frameworks* de atividades de *Red Teaming* que empregam o auxílio de LLMs em contextos adversariais com outro LLMs alvo. A comparação é parametrizada pelo sucesso de *jailbreaks* causados a entidade alvo, além do LLM de ataque. Como contribuição, tem-se o levantamento de sistemático de ferramentas de LLM adversarial no contexto de *Red Team*, bem como uma análise comparativa inicial.

Este artigo é organizado como segue. Inicialmente, a Seção 2 faz uma breve revisão dos conceitos de IA e sua relação com aspectos de cibersegurança. A Seção 3 traz a revisão sistemática realizada, seguida de uma análise comparativa inicial na Seção 4. Por fim, a Seção 5 apresenta as considerações e trabalhos futuros.

2 IA e Cibersegurança

Os LLMs (Figura 1) são modelos de IA do tipo generativo que processam texto através de redes neurais profundas, *Generative Deep Neural Networks* (GDNN), no qual, após o treino para entendimento de contexto desses dados de entrada, são gerados novos dados com o intuito de completar a próxima sentença mais provável [11].

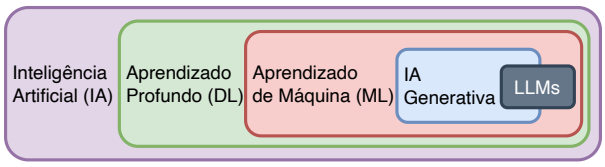

Figura 1. Relação entre IA até LLMs. Adaptado de [12].

Dentro desse tipo de modelo, textos são compreendidos de forma diferente do que por humanos, pois não há entendimento de frases, e sim unidades de texto chamadas *tokens*, que podem ser parte de palavras ou até mesmo mais de um conjunto de palavras, e estas são ordenados em um espaço chamado *embedding space*.

Através do GPT, foi possível aumentar a escala de processamento e entendimento de contexto destes algoritmos, com uma arquitetura de rede neural chamada de *Transformer*, capaz de utilizar um bilhão de *tokens* em treinamento [13]. Na fase de treino desses modelos, são usados cerca de um milhão de parâmetros, no qual o algoritmo tenta representar *tokens* em um espaço multidimensional para a representação de contexto a partir da distância desses elementos em n dimensões [2]. Um algoritmo, para ser categorizado como LLM, deve conseguir aplicar uma regressão probabilística de *tokens* com dados de treinamento para uma entrada de um usuário, além de ter um volume de parâmetros alto e também de *tokens* [14].

A fim de entender vulnerabilidades dessas aplicações, é preciso entender seus recortes e características dentro de cada segmento. Para separar a diversidade de algoritmos neste grupo, dentre as LLMs, e além de algoritmos GPT, o surgimento destes entre diversas organizações e suas variações dentro destes projetos é uma forma de recorte entre os modelos como forma de classificação. Exemplos disso são *Bidirectional Encoder Representations from Transformers* (BERT) desenvolvido pela Google [15], e suas variações, e algoritmos com arquiteturas *Compute-Optimal*, em que está o algoritmo *Large Language Model Meta AI* (LLaMa), mantido pela Meta como código aberto [16].

Há também classificações sobre o tipo de acesso a arquitetura da aplicação: modelos *white-box* e *black-box*. No *white-box* estão arquiteturas em que se tem total acesso a aplicação, como código fonte e com acesso aos dados de treinamento. Assim, o atacante tem total acesso a arquitetura da aplicação, seus parâmetros, e dados de treino [17]. Já em modelos *black-box*, não há entendimento sobre esses aspectos citados, como é o caso de LLMs dentro do grupo GPT operados pela OpenAI, que apenas disponibiliza o código fonte do seu modelo antigo, o GPT-2, além de sua ferramenta de avaliação de referência em modelos do tipo GPT-4, como visto no repositório da empresa no Github [18] e *website* [19]. O completo entendimento e previsibilidade dos resultados desses modelos não é possível dado sua natureza, mesmo quando se trata de aplicações *white-box*. Isso se deve ao fato destes modelos com redes neurais utilizarem de probabilidade em conjunto com uma larga escala de parâmetros, elevando a complexidade desse algoritmos, resultando na dificuldade de entender todas as suas possibilidades. Dadas estas classificações e diversidade de modelos, os LLMs possuem vulnerabilidades que ainda precisam ser exploradas, e apesar de trazerem resultados de texto com velocidade expressiva, não há previsibilidade de seus riscos de forma completa, e são sensíveis a vários tipos de vieses. Exemplificando, LLMs possuem um risco substancial de mensagens contendo insultos, como estereótipos, ideologia extremista, e misoginia a depender de dados de treinamento com os mesmos problemas [7].

Desde a base de treinamento, contendo dados problemáticos levando a respostas ofensivas ou com dados sigilosos, até vazamento de informações através de *prompt injection*, como mostram ferramentas que encontram *jailbreaks* nestes modelos[20]. Sob o aspecto legal, o entendimento sobre a regulamentação das aplicações que usam IA generativo tem sido destaque, e.g., no governo brasileiro [21]. Isso inclui, desde o aspecto de proteção aos dados sensíveis sobre privacidade, até os problemas de propriedade intelectual. As possíveis vulnerabilidades dentro destas aplicações são uma preocupação atual [21].

Em aplicações que utilizam LLMs, há ataques são focados a nível de manipulação de *prompts*, e estes buscam quebrar a linha de entendimento do LLM atacado [2]. Outras frentes são alinhadas a ataques no nível de servidor e o *back-end* da aplicação, afetando as fontes de dados privados, por exemplo. Com isso, a área de segurança da informação pode disponibilizar métodos de garantir a integridade de dados, desenvolvendo testes de vulnerabilidades próprios para diferentes cenários, inclusive sobre LLMs.

Uma abordagem de cibersegurança para proteção de dados é através de times especializados, sendo popular a classificação de: técnicas defensivas como *Blue Teaming*; manipulação de ataques controlados, através do *Red Teaming*; ou a mescla destas abordagens pelo *Purple Teaming*. O foco neste artigo se concentra na visão de comparação de atividades *Red Teaming*, e esta exige a simulação de cenários controlados de ataques contra uma aplicação. Na área de IA pode-se aplicar essas atividades em três principais abordagens, principalmente analisando comportamentos prejudiciais dentro dos LLMs [17]:

1. *Jailbreaking*: anula ou ignora restrições em uma IA generativa.
2. Ataques Adversários: foca em dados de entrada modificados intencionalmente para produzir saídas erradas.
3. *Prompt Injection*: similar a ataques do tipo SQL *Injection*, visando entradas de texto com partes maliciosas.

A fim de achar estas vulnerabilidades descritas, o campo de segurança adota também uma técnica já antes utilizada sobre modelos de IA: a arquitetura de modelo adversarial. Nesse cenário, há dois LLMs que visam aperfeiçoar mutuamente através de *feedbacks* entre si, como os usados em algoritmos de IA sobre classificação de elementos gráficos. Ao ser implementado em contexto de refinamento sobre ataques em LLMs, um algoritmo alvo e um adversarial são responsáveis pela evolução de *prompts* de ataques capazes de obterem *jailbreaks* destes modelos. Atualmente, há ferramentas especializadas neste tipo de cenário, e a comparação destas para testes sobre vulnerabilidades em LLMs é relevante.

3 Trabalhos relacionados

Neste trabalho, foi realizada uma revisão sistemática sobre ferramentas adversariais no contexto de LLMs, com um modelo alvo e outro de ataque, a fim de identificar comparações entre estas ferramentas e também as principais vulnerabilidades expostas nos LLMs. Neste estudo foram analisadas as seguintes questões de pesquisa (RQ) a serem respondidas.

- RQ1. Qual é o número de estudos publicados sobre *Red Teaming* em LLM utilizando ferramentas com LLM no período de 2022 a 2024?
- RQ2. Quais são os critérios de exclusão sobre os artigos selecionados pela busca inicial?
- RQ3. Quais são os critérios de avaliação sobre o sucesso de *jailbreaks* das ferramentas analisadas?
- RQ4. Quais são outros critérios de avaliação na entidade alvo?
- RQ5. Quais são os LLMs adversariais analisados?
- RQ6. Qual ferramenta se mostra com maior taxa de sucesso pelos critérios de avaliação propostos?

Dentro do contexto de pesquisa da área de LLM, artigos recentes são foco na parte desta revisão sistemática, dado que uma das ferramentas de estado da arte sobre ataques em LLMs, que serve de base de comparações nesse trabalho, é de 2023 [20]. Portanto, é natural que um critério de exclusão nas buscas seja de referências posteriores a 2023, além de ferramentas de código aberto. Os artigos os quais não foi possível ter acesso (aberto ou via Portal de Periódicos da Capes) também foram descartados, além da exclusão de artigos que não tinham foco em apresentar uma ferramenta auxiliada por LLMs para atividades de *Red Team* em LLMs. As chaves de busca utilizadas foram: *LLM*, *Red Teaming* ou *Red Team* e *adversarial*. A base de dados explorada foi o Google Scholar, na qual a chave de busca utilizada foi: "*red-teaming and (llm or llms or large-language-models) and adversarial and (jailbreak or framework)*". Foram acrescentados os termos "*-survey*" e " *- benchmark*" na *string* de pesquisa final para evitar resultados irrelevantes de acordo com os critérios de exclusão. Um total de 69 artigos foram encontrados (Tabela 1), levando em conta o filtro de data de publicação de 2022 a 2024, sendo a última pesquisa de atualização realizada 03 de Outubro de 2024.

Tabela 1. Resultados nas bases pesquisadas.

Base pesquisada	Artigos identificados	
	2023	2024
Arxiv	11	37
IEEE Xplore	1	2
Springer	0	4
OpenOverview	1	2
AAAI	0	1
Aclanthology	0	1
HDSR	0	1
Kluedo	0	1
MIT Lib	0	1
Oxford Academic	0	1
RAND	0	1
ResearchGate	1	1
SDJT	0	1
UCL	0	1
Repo. UWAI	0	1

Com estes artigos, foram analisados os resumos e escopo do trabalho para avaliar sua adequação a revisão sistemática, e os termos de exclusão acima citados foram os critérios de filtro na análise qualitativa. Pesquisas sobre ferramentas de defesa, mesmo que alinhadas a *Red Teaming* foram dispensadas, além de análises bibliográficas sobre a área de vulnerabilidades de *Red Teaming* e outras ferramentas fora do escopo, como o caso de algoritmos que não usam LLM como auxílio para *jailbreaking*. Após a implicação destes critérios, 10 artigos foram selecionados para a revisão, nos quais há presença de um *framework* de código aberto, que utiliza de LLMs para construção de ataques *jailbreak* no cenário de *prompts* de LLMs. A abrangência sobre transferibilidade de ataques de ferramentas entre diversos tipos de LLMs como alvo também pode ser critério para trabalhos futuros. O objetivo principal das aplicações encontradas é produzir *jailbreaks* e analisar sua taxa de acerto. A forma como a análise de taxa de acerto para cada artigo foi de analisar se há palavras ofensivas baseado em um banco de dados [22]. Após a revisão bibliográfica, a análise de outras bibliografias base sobre o estado da arte referenciadas nos trabalhos também incluiu o *framework Greedy Coordinate Gradient* (GCG) [20].

4 Análise comparativa

Com a seleção de artigos pelos critérios avaliados, a comparação entre trabalhos foi realizada de forma primária pelos próprios autores dos algoritmos, com os parâmetros pelo ponto de vista do algoritmo de estado da arte citado em mais de um trabalho, vulgo GCG. Os artigos que citam e comparam seus *frameworks* ao anteriormente citado são:

- AgentPoison [23]: ferramente de ataque contra LLMs genéricos, envenenando sua memória de longo prazo.
- Fuzz Testing-Driven [24]: uma nova estrutura de *jailbreaks* automatizada, que utiliza de estratégia de teste de Fuzzy com projeto de ataques personalizados.
- EnJa [25]: apresenta o conceito de Ensemble Jailbreak (EnJa) e explora métodos que podem integrar *jailbreak* de nível de *prompt* e nível de *token* em um ataque híbrido mais poderoso.
- AutoDAN [26]: utilizando algoritmo genético hierárquico, AutoDan consegue gerar *prompts* com capacidade de *jailbreaks* de forma automatizada.
- ToxDet [27]: uma ferramenta que provoca o modelo alvo por um LLM para produzir respostas tóxicas utilizando aprendizado por reforço para otimização.
- DeGCG [28]: aprendizagem de duas etapas sobre *tokens* responsáveis por desbloquear comportamentos nocivos em LLMs.
- ReNeLLM [29]: generaliza ataques de *prompt* de LLMs em dois tipos (i-reescrita e ii-de aninhamento de cenário) e propõe um novo *framework* capaz de desbloquear essas vulnerabilidades de forma automática com LLMs.

Analisando pela perspectiva do LLM alvo, todos os estudos comparados resultantes mostraram experimentos em relação ao modelo LLaMa, fazendo assim uma fonte de métrica relevante. Além disso, mais de um trabalho referencia comparações entre outros *frameworks*, sendo outra fonte de métrica. Desta forma, há quatro estudos resultantes, sendo: ReNeLLM, DeGCG, EnJa e AgentPosition.

O algoritmo AgentPoison [23] não atende a todos os critérios ao especificar uma ferramenta como alvo, ao contrário das outras aplicações que agem diretamente em um LLM alvo. Outra comparação em relação ao modelo de estado da arte é feita pela ferramenta EnJa [25], no qual o modelo de ataque de código aberto usado é o Vicuna (Vicuna-7B e Vicuna-13-v1.5), como modo de comparação, além de LLaMa. O modelo Vicuna é de código aberto, treinado pelo ajuste fino do LLaMA em conversas compartilhadas por usuários coletadas do ShareGPT [30]. O LLaMa, da empresa Meta, é um modelo de estado da arte dos LLMs de código aberto. Isto mostra que em um intervalo de um ano, há novas ferramentas que mostram evidências de que ultrapassam um modelo referência na área de LLMs. A *Taxa Média de Sucesso* (TMS), Tabela 2, foi uma métrica comum entre os estudos, apesar de ser dada em porcentagem.

Tabela 2. Comparativo TMS: algoritmo *Ensemble Jailbreak Framework* (EnJa).

Ferramenta	LLM do lado Atacante	TMS da LLM Alvo (%)		
		Vicuna-7B	Vicuna-13B	Llama-2-13B
GCG	Vicuna-7B	86.0	88.0	38.0
Enja	Vicuna-13B-v1.5	98.0	98.0	94.0

A TMS utilizada provém de um modelo de LLM usado como juiz, e.g., Vicuna-13B neste estudo, de forma que a taxa pode ter parâmetros não padronizados para uma comparação ampla. Com resultados nesta comparação, o algoritmo GCG diminui sua TMS com o modelo Llama2-13B, ao contrário do EnJa, que continua com taxas acima de 90%, garantindo sua transferibilidade. Por não usarem o mesmo modelo de LLM atacante, ainda é necessário uma reprodução desta comparação, de forma a garantir sua veracidade, e confirmar os resultados obtidos.

Os programas também não estavam em um mesmo ambiente de desenvolvimento, como o algoritmo EnJa, que foi testado em um ambiente que detinha recursos de hardware com uma GPU NVIDIA A100, enquanto o modelo DeGCG foi implementado em uma GPT NVIDIA A40. Outra forma proposta, com a divisão de cenários aplicados (Tabela 3), é a de contrapor a taxa de acerto entre os dois modelos, para avaliar a evolução entre os modelos em relação com GCG, com o mesmo LLM alvo nos dois cenários.

Tabela 3. Comparativa TMS: algoritmo *Prompt **Re**writing and Scenario **Ne**sting **LMM** Framework* (ReNeLLM).

Cenário	Ferramenta	TMS (%) LLM Alvo (LLaMa-2-7b-chat)
Método de reescrita com GPT-4	ReNeLLM	100.0
	GCG	32.1
Llama2-chat usado como LLM A	DeGCG	43.9
	GCG	21.7

No primeiro modelo, a TMS avalia um método de reescrita aliado ao modelo GPT-4, em que esse sistema de avaliação também utiliza de um LLM em uma das suas duas formas de verificação. No modelo DeGCG, o processo conta com mais de um modelo de LLM de avaliação. Dado estas diversas formas de avaliação, apesar de terem bases parecidas, a comparação direta não é totalmente adequada. Ao usar dados resultantes de experimentos com diferentes parâmetros de comparação entre *frameworks*, ambientes computacionais variados, e diferentes LLMs de ataque e de alvo, não há garantia sobre a manutenção deste estado ao colocar essas aplicações em cenários iguais para novas métricas.

A reprodutibilidade é necessária para avaliar uma comparação direta entre ReNeLLM e DeGCG. As condições de hardware iguais para testes além de assegurar a imparcialidade entre comparações, pode demonstrar e reiterar os dados mostrados nos estudos originais para novos trabalhos da área. As condições de quantidade de tentativas de consultas realizadas por cada ferramenta para a atividade de *prompt injection* também deve ser padronizada para assegurar os resultados finais. Contudo, ainda é possível dizer que há evoluções em relação ao método CGC, por tanto ReNeLLM, DeGCG e EnJa mostrarem suas taxas mais elevadas que o modelo de referência em seus estudos.

5 Considerações e trabalhos futuros

A análise comparativa foi realizada dada a taxa de sucesso indicada por cada trabalho analisado. Tendo em vista os diferentes tipos de LLMs usados para tanto modelo de ataque quanto de alvo, a comparação foi realizada com comparações que correspondiam os mesmo cenários. Uma análise sobre todas as ferramentas selecionadas neste trabalho em uma mesma comparação é visada em estudos futuros, respeitando critérios de avaliação, considerando as mesmas entidades alvo, além de métricas sobre o mesmo elementos de hardware.

Para futuros trabalhos é pretendido que, além de outras revisões sistemáticas mais abrangentes, faça-se uso de testes empíricos com os modelos que estiverem disponíveis, para assegurar a reprodutibilidade dos trabalhos mencionados, como para por em igualdade o ambiente e configurações para a adequada comparação entre LLMs.

Agradecimentos:
Os autores agradecem o apoio do LARC/USP, LabP2D/UDESC, FDTE e FAPESC. Este trabalho foi apoiado pelo CNPq (processos 307732/2023-1 e 311245/2021-8), FAPESP (processo 2020/09850-0) e CAPES (Código de Financiamento 001).

Referências

1 Hadi, M. U. *et al. A Survey on Large Language Models: Applications, Challenges, Limitations, and Practical Usage.* 10 jul. 2023. DOI: 10.36227/techrxiv.23589741.v1. Disponível em: https://www.techrxiv.org/doi/full/10.36227/techrxiv.23589741.v1. Acesso em: 14 set. 2024.

2 Kucharavy, A. *et al.* Large Language Models in Cybersecurity: Threats, Exposure and Mitigation. *In*: 2024. P. 247. DOI: 10.1007/978-3-031-54827-7.

3 Woldseth, R. V. *et al.* On the use of artificial neural networks in topology optimisation. *Structural and Multidisciplinary Optimization*, Springer Science e Business Media LLC, v. 65, n. 10, out. 2022. ISSN 1615-1488. DOI: 10.1007/s00158-022-03347-1. Disponível em: http://dx.doi.org/10.1007/s00158-022-03347-1.

4 AI Chatbot for Banking - IBM Watsonx Assistant. Disponível em: https://www.ibm.com/br-pt/products/watsonx-assistant/banking. Acesso em: 14 set. 2024.

5 BB usa tecnologia generativa para apoiar atendimento. Disponível em: https://www.bb.com.br/pbb/pagina-inicial/imprensa/n/67461/bb-usa-tecnologia-generativa-para-apoiar-atendimento#/. Acesso em: 14 set. 2024.

6 BIA Bardesco Inetligência Artificial. Disponível em: https://banco.bradesco/bia/. Acesso em: 14 set. 2024.

7 Bender, E. *et al.* On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? *In*: p. 610–623. DOI: 10.1145/3442188.3445922.

8 Liu, X. *et al. Robustifying Safety-Aligned Large Language Models through Clean Data Curation.* 2024. arXiv: 2405.19358 [cs.CR]. Disponível em: https://arxiv.org/abs/2405.19358.

9 Huang, J.; Shao, H.; Chang, K. C.-C. *Are Large Pre-Trained Language Models Leaking Your Personal Information?* 2022. arXiv: 2205.12628 [cs.CL]. Disponível em: https://arxiv.org/abs/2205.12628.

10 Kumar, P. Adversarial attacks and defenses for large language models LLMs: methods, frameworks and challenges. *Int J Multimed Info Retr 13, 26*, 2024. DOI: https://doi.org/10.1007/s13735-024-00334-8.

11 Oussidi, A.; Elhassouny, A. Deep generative models: Survey. *In*: 2018 International Conference on Intelligent Systems and Computer Vision (ISCV). 2018. P. 1–8. DOI: 10.1109/ISACV.2018.8354080.

12 Shahab, O. *et al.* Large language models: a primer and gastroenterology applications. *Therapeutic Advances in Gastroenterology*, v. 17, fev. 2024. DOI: 10.1177/17562848241227031.

13 Vaswani, A. *et al. Attention Is All You Need.* 2023. arXiv: 1706.03762 [cs.CL]. Disponível em: https://arxiv.org/abs/1706.03762.

14 Kaplan, J. *et al. Scaling Laws for Neural Language Models.* 2020. arXiv: 2001.08361 [cs.LG]. Disponível em: https://arxiv.org/abs/2001.08361.

15 Devlin, J. *et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* 2019. arXiv: 1810.04805 [cs.CL]. Disponível em: https://arxiv.org/abs/1810.04805.

16 LLAMA 3.2. Disponível em: https://www.llama.com. Acesso em: 14 set. 2024.

17 Lin, L. *et al. Against The Achilles' Heel: A Survey on Red Teaming for Generative Models.* 2024. arXiv: 2404.00629 [cs.CL]. Disponível em: https://arxiv.org/abs/2404.00629.

18 GITHUB Openai. Disponível em: https://github.com/openai. Acesso em: 14 set. 2024.

19 OPENAI GPT4 Research. Disponível em: https://openai.com/index/gpt-4-research. Acesso em: 14 set. 2024.

20 Zou, A. *et al. Universal and Transferable Adversarial Attacks on Aligned Language Models.* 2023. arXiv: 2307.15043 [cs.CL]. Disponível em: https://arxiv.org/abs/2307.15043.

21 Senado, A. Relator apresenta relatório atualizado sobre regulamentação da IA. *Senado Notícias*, 20024. Disponível em: https://www12.senado.leg.br/noticias/materias/2024/07/04/relator-apresenta-relatorio-atualizado-sobre-regulamentacao-da-ia.

22 Jiang, B. *et al. DART: Deep Adversarial Automated Red Teaming for LLM Safety.* 2024. arXiv: 2407.03876 [cs.CR]. Disponível em: https://arxiv.org/abs/2407.03876.

23 Chen, Z. *et al. AgentPoison: Red-teaming LLM Agents via Poisoning Memory or Knowledge Bases.* 2024. arXiv: 2407.12784 [cs.LG]. Disponível em: https://arxiv.org/abs/2407.12784.

24 Gong, X. *et al. Effective and Evasive Fuzz Testing-Driven Jailbreaking Attacks against LLMs.* 2024. arXiv: 2409.14866 [cs.CR]. Disponível em: https://arxiv.org/abs/2409.14866.

25 Zhang, J. *et al. EnJa: Ensemble Jailbreak on Large Language Models.* 2024. arXiv: 2408.03603 [cs.CR]. Disponível em: https://arxiv.org/abs/2408.03603.

26 Liu, X. *et al. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models.* 2024. arXiv: 2310.04451 [cs.CL]. Disponível em: https://arxiv.org/abs/2310.04451.

27 Du, Y. *et al. Detecting AI Flaws: Target-Driven Attacks on Internal Faults in Language Models.* 2024. arXiv: 2408.14853 [cs.CL]. Disponível em: https://arxiv.org/abs/2408.14853.

28 Liu, H. *et al. Advancing Adversarial Suffix Transfer Learning on Aligned Large Language Models.* 2024. arXiv: 2408.14866 [cs.CL]. Disponível em: https://arxiv.org/abs/2408.14866.

29 Ding, P. *et al. A Wolf in Sheep's Clothing: Generalized Nested Jailbreak Prompts can Fool Large Language Models Easily.* 2024. arXiv: 2311.08268 [cs.CL]. Disponível em: https://arxiv.org/abs/2311.08268.

30 VICUNA, An Opensource Chatbot Impressing GPT-4 with 90% ChatGPT Quality. Disponível em: https://lmsys.org/blog/2023-03-30-vicuna. Acesso em: 14 set. 2024.