UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
CIÊNCIA DA COMPUTAÇÃO

EDUARDO ALEXANDRE DE AMORIM

SecBERT: Aprimorando a Segurança de LLMs em Português via Detecção de Jailbreak Prompts

Recife

2025

EDUARDO ALEXANDRE DE AMORIM

SecBERT: Aprimorando a Segurança de LLMs em Português via Detecção de Jailbreak Prompts

Trabalho de Conclusão de Curso apresentado no curso de Bacharelado em Ciência de Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Cientista da Computação.

Orientador: Cleber Zanchettin

Recife

2025

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Amorim, Eduardo Alexandre de.
SecBERT: Aprimorando a Segurança de LLMs em Português via Detecção de Jailbreak Prompts / Eduardo Alexandre de Amorim. - Recife, 2025.
40 p : il., tab.

Orientador(a): Cleber Zanchettin
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Ciências da Computação - Bacharelado, 2025.
Inclui referências, apêndices, anexos.

1. Segurança em LLMs. 2. Jailbreak Prompts. 3. PLN. 4. Classificação de Texto. I. Zanchettin, Cleber. (Orientação). II. Título.

000 CDD (22.ed.)

EDUARDO ALEXANDRE DE AMORIM

SecBERT: Aprimorando a Segurança de LLMs em Português via Detecção de Jailbreak Prompts

Trabalho de Conclusão de Curso apresentado no curso de Bacharelado em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em: 05/08/2025

BANCA EXAMINADORA

Profº. Dr. Cleber Zanchettin(Orientador)
Universidade Federal de Pernambuco

Profº. Dr. Giordano Ribeiro Eulalio Cabral (Examinador Interno)
Universidade Federal de Pernambuco

À minha família, que sempre acreditou em mim e foi meu alicerce em cada passo dessa jornada. Sou imensamente grato por todo o apoio, amor e incentivo incondicional.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, por me conceder diariamente saúde, força e coragem para seguir em frente. Nos momentos mais difíceis, quando as dúvidas e os medos pareciam maiores que as minhas forças, foi Ele quem me sustentou. Sua presença constante me deu direção, clareza e fé para continuar, mesmo quando tudo parecia incerto. Sem Deus, essa caminhada não teria sido possível. A Ele, toda a minha gratidão por cada aprendizado, cada superação e cada conquista.

À minha mãe, mulher forte e incansável, que sempre abdicou de tantas coisas por mim. Seu amor incondicional, sua coragem diante das adversidades e sua capacidade de me levantar foram fundamentais para que eu chegasse até aqui. Em muitas ocasiões, quando tudo parecia desmoronar, sua presença firme e seu olhar de confiança foram o que me fizeram continuar. Cada conquista minha carrega um pedaço seu, e sou eternamente grato por tudo que você fez, e faz, por mim.

Ao meu pai, que sempre foi um exemplo de dedicação, inteligência e compromisso. Seu incentivo aos estudos e à busca pelo conhecimento sempre foram uma grande inspiração. Sua forma de se preocupar, de orientar com sabedoria e, ao mesmo tempo, de demonstrar amor silenciosamente, me ensinou muito sobre o que é ser responsável, comprometido e íntegro. Obrigado por acreditar em mim em todos os momentos.

À minha esposa, Thamires, que trouxe luz para minha vida e me mostrou um novo sentido para cada conquista. Sua presença constante, seu apoio nos dias bons e ruins, sua paciência, carinho e compreensão foram essenciais para que eu mantivesse o equilíbrio. Você fortalece quando eu estou, me incentiva quando quero desistir, e me faz lembrar todos os dias por que vale a pena lutar pelos meus sonhos. Essa vitória é nossa.

Ao meu irmão João, que desde pequeno tem sido meu companheiro e fonte de inspiração. Mesmo mais novo, ele me ensina diariamente o valor da simplicidade, da empatia e da alegria genuína. Seu jeito de ver o mundo me faz refletir e querer ser uma pessoa melhor. Ter você por perto torna a vida mais leve e mais bonita. Essa conquista também é sua, irmão. Obrigado por tudo.

RESUMO

O crescimento dos Modelos de Linguagem de Grande Escala (LLMs) traz desafios à segurança, sobretudo diante do uso de *jailbreak prompts*, instruções criadas para burlar salvaguardas. Embora o tema esteja em debate na literatura internacional, há escassez de soluções voltadas à língua portuguesa. Este trabalho propõe o SecBERT, um classificador treinado para detectar *jailbreaks* em português. Para isso, adaptou-se o *WildJailbreak Dataset* via tradução automatizada, resultando em 29.432 exemplos rotulados em quatro categorias. Foram conduzidos alguns experimentos com modelos BERT (e.g. BERTimbau, RoBERTa), testando diferentes estratégias de *fine-tuning*. Os resultados evidenciam que modelos ajustados ao idioma superam abordagens multilíngues ou generalistas. O SecBERT representa, portanto, um avanço na segurança de LLMs em português.

Palavras-chave: Segurança em LLMs; *Jailbreak Prompts*; PLN; Classificação de Texto.

ABSTRACT

The rise of Large Language Models (LLMs) poses security challenges, especially in the face of jailbreak prompts—crafted instructions designed to bypass safeguards. While the topic is under discussion in the international literature, there is a lack of solutions tailored to the Portuguese language. This work proposes SecBERT, a classifier trained to detect jailbreaks in Portuguese. To this end, the *WildJailbreak Dataset* was adapted via automated translation, resulting in 29,432 labeled examples across four categories. Several experiments were conducted using BERT-based models (e.g., BERTimbau, RoBERTa), testing different fine-tuning strategies. Results show that language-specific models outperform multilingual or general-purpose approaches. SecBERT thus represents a step forward in securing LLMs in Portuguese.

Keywords: LLM Security; *Jailbreak Prompts*; NLP; Text Classification.

LISTA DE FIGURAS

1 Exemplo da visualização do teste de KS 14
2 Arquitetura geral 21
3 Teste do KS do BERTimbau-base com Backbone descongelado. 24
4 Curva ROC do BERTimbau-base com Backbone descongelado. 24
5 Teste do KS do BERTimbau-large. 26
6 Curva ROC do BERTimbau-large. 26
7 Teste do KS do BERT Base Uncased. 28
8 Curva ROC do BERT Base Uncased. 28
9 Teste do KS do RoBERTa-base. 30
10 Curva ROC do RoBERTa-base. 30

LISTA DE TABELAS

1 Comparativo entre tipos de *Prompt Hacking* em LLMs 12
2 Categorias do dataset *WildJailbreak* 15
3 Campos presentes no dataset *WildJailbreak* 16
4 Distribuição percentual dos registros por tipo de dado no dataset *WildJailbreak* 16
5 Similaridade de cosseno entre traduções geradas por `GPT-4o-mini` e `GPT-4o`, por categoria 18
6 Divisão do dataset para os experimentos 18
7 Mapeamento de `data_type` para classe binária 19
8 Baseline 22
9 BERTimbau-base com Backbone descongelado 23
10 BERTimbau-large 25
11 BERT Base Uncased 27
12 RoBERTa-base 29
13 Resumo Comparativo dos Experimentos 31

Sumário

1 INTRODUÇÃO **8**

1.1 Contexto 8

1.2 Motivação 9

1.3 Objetivos da pesquisa 9

1.4 Limitações 10

2 REFERENCIAL TEÓRICO **11**

2.1 Tipos de *Prompt* Hacking 11

2.2 Teste de Kolmogorov–Smirnov (KS) 13

3 METODOLOGIA **15**

3.1 Criação do dataset 15

3.2 Divisão para experimento 18

3.3 Arquitetura do modelo e configuração dos experimentos 19

4 RESULTADOS **22**

4.1 Baseline 22

4.2 BERTimbau-base com Backbone descongelado 23

4.3 BERTimbau-large 25

4.4 BERT Base Uncased 27

4.5 RoBERTa-base 29

4.6 Síntese Comparativa 31

4.7 Modelo escolhido para o SecBERT e Implicações Práticas 32

5 CONCLUSÃO **33**

5.1 Trabalhos Futuros 34

5.2 Considerações Finais 34

Referências **36**

1 INTRODUÇÃO

1.1 Contexto

A crescente utilização global dos Modelos de Linguagem de Grande Escala (LLMs) e suas amplas capacidades em diversos domínios têm transformado a interação humana com a inteligência artificial. Contudo, paralelamente ao seu potencial impressionante, a sua utilização indevida tem suscitado preocupações significativas. Incidentes recentes evidenciam os riscos de LLMs gerarem desinformação, promoverem teorias da conspiração, escalarem ataques de phishing e facilitarem campanhas de ódio. Além disso, estudos como já apontam a alavancagem contínua de LLMs de mercado, como os modelos disponibilizados pela Openai via ChatGPT, para atividades cibercriminosas (RABABAH et al., 2024; GUPTA et al. 2023)

Diante desses riscos, uma categoria particular de prompts adversários, conhecidos como *jailbreak prompts*, emergiu como o principal vetor de ataque para contornar as salvaguardas dos LLMs e, assim, extrair conteúdo prejudicial. Esses prompts são intencionalmente elaborados para manipular os LLMs a gerar conteúdo que viola as políticas de uso estabelecidas pelos fornecedores. Um exemplo notório é a capacidade de um *jailbreaks prompt* fazer com que um LLM gere conteúdo restrito, impróprio ou perigoso, mesmo quando o modelo recusaria apropriadamente a mesma pergunta sem o prompt adversário . Tais prompts são considerados de altíssimo risco, comprometendo a integridade e a ética do modelo (SHEN et al., 2024; RABABAH et al., 2024)

A comunidade de pesquisa ainda carece de uma compreensão sistemática dos *jailbreak prompts*, tanto das suas características e padrões de evolução, como a extensão dos danos que podem causar . Eles tendem a ser significativamente mais longos, com uma média de **555 *tokens*** (1,5 vezes o comprimento de prompts regulares), recorrendo a diversas estratégias de *jailbreak*, como ***roleplaying* (interpretação de papéis), virtualização, simulação de cenários alternativos e aumento de complexidade semântica**. A explicação detalhada das possíveis categorias propostas de prompt hacking no (RABABAH et al., 2024) estão na subseção 2.1 (SHEN et al., 2024; RABABAH et al., 2024)

Apesar de fornecedores de LLMs, como a OpenAI, implementarem salvaguardas (e.g., *reinforcement learning from human feedback* — RLHF) para alinhar os modelos com valo-

res humanos, diversas estratégias de defesa adicionais, como filtros de entrada/saída, vêm sendo empregadas para aprimorar a segurança e a robustez dos LLMs frente a ataques *jailbreak*. Segundo a literatura, essas abordagens colaboram para melhorar a capacidade dos modelos de reconhecer e rejeitar prompts maliciosos, contribuindo diretamente para o aumento da confiabilidade e segurança dos sistemas (CHRISTIANO et al., 2022; RABABAH et al., 2024).

1.2 Motivação

Embora as pesquisas sobre defesas contra *jailbreak prompts* estejam em expansão, ainda há uma lacuna significativa na compreensão sistemática dessas ameaças e na criação de defesas eficazes. Essa escassez é ainda mais evidente no contexto da língua portuguesa. Apesar da existência de modelos como o BERTimbau, voltados para tarefas de PLN em português brasileiro, há poucos estudos sobre a segurança de LLMs nesse idioma. Estratégias desenvolvidas para o inglês nem sempre se aplicam diretamente a outras línguas, devido a particularidades linguísticas e culturais, exigindo abordagens específicas. Diante disso, este trabalho propõe o desenvolvimento do SecBERT, um modelo voltado à detecção de *jailbreak prompts* em português, contribuindo para o fortalecimento das salvaguardas de LLMs usados no Brasil e em outros países lusófonos (SOUZA et al. 2020).

1.3 Objetivos da pesquisa

Este trabalho tem como objetivo principal desenvolver e avaliar o *SecBERT*, um modelo especializado na detecção de *jailbreak prompts* em Modelos de Linguagem de Grande Escala (LLMs) que operam em português, visando aprimorar a segurança e a robustez desses sistemas. Mais especificamente, busca-se:

- Adaptar de parte do conjunto de dados fornecido pelo *WildJailbreak Dataset* para o português, com o auxílio de modelos de tradução automática em lote (DONG et al. 2024);
- Desenvolver o modelo *SecBERT*, experimentando diferentes arquiteturas baseadas em BERT, assim como diversas estratégias de pré-processamento e treinamento supervisionado;

- Analisar as métricas de performance do *SecBERT* na detecção de *jailbreak prompts* em diferentes cenários, contra diversas estratégias de ataque disponibilizadas no *WildJailbreak* , com base em métricas clássicas de classificação binária, como KS e AUC-ROC (DONG et al. 2024);

- Propor diretrizes para a melhoria contínua das salvaguardas de LLMs em português, com base nos achados e na performance do *SecBERT*, contribuindo para o ecossistema de segurança de IA aplicado a países lusófonos.

1.4 Limitações

Algumas limitações devem ser consideradas quanto à generalização dos resultados obtidos neste trabalho. A adaptação de *prompts* para o português foi realizada por meio de tradução automática, no entanto, estudos como demonstram que esse processo pode comprometer a intenção adversarial original. Em certos idiomas menos representados, como o letão, a tradução direta transformou *prompts* maliciosos em inofensivos, levantando dúvidas sobre a preservação da intenção em diferentes contextos linguísticos. Apesar disso, o mesmo estudo ressalta que a vulnerabilidade a *jailbreaks* multilingues ainda persiste, reforçando a importância de avaliações específicas em cada idioma (KANEPAJS et al., 2024).

Além disso, a natureza dinâmica dos *jailbreak prompts*, com estratégias em constante evolução, exige que modelos de detecção passem por atualizações contínuas. Um sistema eficaz hoje pode se tornar obsoleto diante de novas técnicas de ataque, demandando um ciclo permanente de aprimoramento.

Este trabalho também se restringe à detecção de *jailbreaks prompts*, uma das modalidades de *prompt hacking* descritas em . Outros ataques, como *prompt injection* (inserção de instruções não confiáveis) e *prompt leaking* (extração de informações sensíveis do modelo), não são abordados diretamente, embora possam ser discutidos de forma complementar. Assim, os resultados aqui apresentados refletem exclusivamente o desempenho do SecBERT frente a *jailbreak prompts* em português e não devem ser generalizados para outros tipos de ataque. Mais informações sobre essas categorias encontram-se na subseção 2.1 (RABABAH et al., 2024).

2 REFERENCIAL TEÓRICO

2.1 Tipos de *Prompt* Hacking

A segurança e a robustez das aplicações baseadas em LLMs enfrentam desafios críticos, sendo os ataques por meio de *prompt hacking* uma das principais ameaças à confiabilidade desses sistemas. Conforme sistematizado por, três categorias principais compõem esse tipo de ataque: ***jailbreaking***, ***injection*** e ***leaking*** (RABABAH et al., 2024).

Embora compartilhem o princípio comum de manipular *prompts* para gerar comportamentos não intencionais, essas categorias apresentam finalidades, técnicas e impactos distintos. O Quadro 1 apresenta um resumo comparativo dessas modalidades, com base na taxonomia proposta pelos autores, em tradução livre adaptada.

Quadro 1: Comparativo entre tipos de *Prompt Hacking* em LLMs

Fonte: Elaboração própria, com base em RABABAH et al. (2024) e GUPTA et al. (2023).

Aspecto	**Jailbreaking**	**Injection**	**Leaking**
Definição	Burlar as diretrizes de segurança e ética do modelo.	Manipular entradas para controlar o comportamento do modelo.	Divulgação não intencional de informações sensíveis.
Objetivo	Gerar respostas restritas.	Alterar a saída do modelo de forma específica, geralmente maliciosa.	Expor o *system prompt* ou informações ocultas.
Técnicas	Criar entradas que contornem salvaguardas, como *roleplaying* ou simulações.	Inserir instruções ou sequências específicas dentro do texto de entrada.	Elaborar prompts para induzir a revelação de informações internas.
Casos de uso	Obter instruções sobre atividades ilegais ou conteúdo impróprio.	Forçar o modelo a produzir respostas direcionadas, mesmo que indevidas.	Fazer o modelo revelar detalhes privados sobre sua configuração.
Resultado	Geração de conteúdo contrário às diretrizes éticas.	Saídas controladas, muitas vezes não autorizadas ou nocivas.	Exposição acidental de informações confidenciais.
Alvo	Restrições de segurança e alinhamento do modelo.	Comportamento geral do modelo.	Informações sensíveis do modelo.
Nível de risco	Muito alto, frequentemente associado a conteúdo nocivo ou ilegal.	Moderado a alto, dependendo da manipulação.	Alto, devido a violações de privacidade.
Impacto	Compromete a integridade e a ética do modelo.	Afeta a confiabilidade das respostas geradas.	Viola a privacidade e a confidencialidade do sistema.

Vale destacar que, embora essas categorias sejam descritas separadamente, ataques reais frequentemente combinam elementos de duas ou mais delas. Por exemplo, um prompt pode empregar estratégias de simulação de papéis (*jailbreaking*), ao mesmo tempo em que injeta comandos ocultos (*injection*) com o intuito de extrair informações sensíveis do modelo (*leaking*). Essa sobreposição torna os ataques mais sofisticados, elevando sua eficácia e dificultando a detecção por mecanismos de defesa.

Neste trabalho, o foco recai exclusivamente sobre prompts do tipo *jailbreak*, por se tratar da categoria mais diretamente associada à geração de conteúdos restritos e de alto risco.

2.2 Teste de Kolmogorov–Smirnov (KS)

O teste de Kolmogorov–Smirnov (KS) é uma medida estatística não paramétrica amplamente utilizada para avaliar a diferença entre duas distribuições acumuladas. No contexto de aprendizado de máquina, especialmente em tarefas de classificação binária, o **KS máximo** é utilizado como métrica para mensurar a separabilidade entre as distribuições de pontuações (ou probabilidades) atribuídas às classes positiva e negativa.

De forma prática, o KS máximo é definido como a maior distância entre as curvas de distribuição acumulada (CDF) dos escores de predição para cada classe. Um valor de KS próximo de 0 indica sobreposição quase total entre as distribuições (i.e., o modelo não está conseguindo distinguir as classes), enquanto um valor próximo de 1 indica alta separabilidade.

No contexto deste trabalho, o teste de KS é empregado para avaliar a eficácia do modelo *SecBERT* em distinguir entre prompts benignos e prompts adversariais do tipo *jailbreak*. Ao comparar as distribuições dos escores preditivos atribuídos a cada tipo de prompt, o KS máximo fornece uma indicação clara de quão bem o modelo está separando as duas classes.

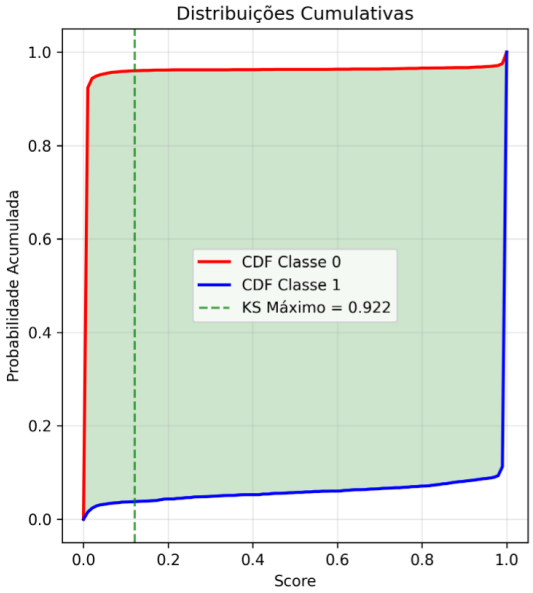
Essa métrica é particularmente relevante quando se deseja avaliar não apenas a acurácia bruta do modelo, mas a sua **capacidade de discriminação**, mesmo em cenários com distribuições desbalanceadas ou sobrepostas.

A Figura 1 apresenta uma ilustração típica do teste de KS, onde as curvas de distribuição acumulada de duas amostras são comparadas. A maior distância vertical entre essas curvas representa o valor de KS máximo, servindo como base para a decisão estatística

sobre a separabilidade das distribuições. Embora a imagem não utilize dados do experimento deste trabalho, ela serve como referência visual para compreender o funcionamento do teste.

Figura 1: Exemplo da visualização do teste de KS

Fonte: Elaboração própria



3 METODOLOGIA

3.1 Criação do dataset

O dataset traduzido utilizado neste estudo foi disponibilizado publicamente no Hugging Face, permitindo reprodutibilidade e experimentações futuras. Ele pode ser acessado através do *link*:

`https://huggingface.co/datasets/Edu-p/jailbreak-prompts-pt`

A construção do dataset utilizado neste estudo baseou-se na adaptação de uma base de dados preexistente, denominada *WildJailbreak*. Esse dataset é estruturado em quatro categorias distintas, com o objetivo de mitigar comportamentos de segurança excessivamente restritivos em LLMs, ao mesmo tempo em que fornece consultas contrastivas tanto requisições prejudiciais (em formatos diretos e adversariais) quanto consultas benignas que imitam a estrutura de *prompts* ofensivos, mas sem conteúdo malicioso (DONG et al. 2024).

As quatro categorias presentes no dataset *WildJailbreak* são apresentadas no Quadro 2:

Quadro 2: Categorias do dataset *WildJailbreak*

Fonte: Elaboração própria, com base em DONG et al. 2024.

Categoria	**Descrição**
Vanilla Harmful	Requisições diretas com potencial de eliciar respostas prejudiciais de Modelos de Linguagem (LMs).
Vanilla Benign	Prompts inofensivos, criados para combater a segurança exagerada, isto é, a recusa excessiva a consultas benignas.
Adversarial Harmful	*Jailbreaks* adversariais que transmitem requisições prejudiciais de forma mais complexa e furtiva.
Adversarial Benign	Consultas adversariais semelhantes a *jailbreaks*, mas sem conter intenção prejudicial.

O dataset original disponibiliza, para cada registro, os campos listados no Quadro 3:

Quadro 3: Campos presentes no dataset *WildJailbreak*

Fonte: Elaboração própria, com base em DONG et al., 2024.

Campo	**Descrição**
`vanilla`	O *prompt vanilla* (pode ser *harmful* ou *benign*).
`adversarial`	O *prompt adversarial* (pode ser *harmful* ou *benign*); o campo é vazio nos exemplos *vanilla*.
`completion`	A resposta do modelo tipicamente uma recusa frente a *prompts* prejudiciais ou uma resposta conforme nos casos benignos.
`data_type`	Indica a categoria do dado: `vanilla_harmful`, `vanilla_benign`, `adversarial_harmful` ou `adversarial_benign`.

Para este estudo, foi realizada uma amostragem aleatória de aproximadamente 10% do volume original do dataset de treino do *WildJailbreak*. Essa seleção resultou em um total de 29.432 registros, com a seguinte distribuição percentual por `data_type`:

Quadro 4: Distribuição percentual dos registros por tipo de dado no dataset *WildJailbreak*

Fonte: Elaboração própria

`data_type`	Porcentagem (%)
`vanilla_benign`	26,65%
`vanilla_harmful`	26,47%
`adversarial_harmful`	24,71%
`adversarial_benign`	22,17%

Todos os registros selecionados foram traduzidos para o português brasileiro por meio de uma abordagem semi-automatizada, utilizando o modelo `GPT-4o-mini` com validação humana posterior. Casos recorrentes de traduções incorretas foram tratados por meio da repetição do processo ou pela exclusão dos registros durante a etapa de consolidação final do corpus. Um exemplo comum de erro de tradução ocorreu quando o retorno do modelo era truncado, geralmente devido ao comprimento excessivo do *prompt* enviado ao provedor.

O *prompt* específico utilizado para tradução seguiu o seguinte formato:

```
Retorne a tradução mais correta fidedigna possível da seguinte tabela para o
português brasileiro.

{tabela}

A estrutura de retorno (json) será a seguinte:

{
"prompt": ,
"data_type": <>,
}
```

O custo total do processo de tradução foi de `$2,79`, o que equivale a aproximadamente $\$1{,}10 \times 10^{-4}$ por registro. Após a conclusão dessa etapa, os *prompts* traduzidos foram agregados, compondo a versão final do dataset.

Durante o pré-processamento do corpus final, foi adicionada uma nova coluna chamada `target`, a qual representa a variável dependente utilizada no processo de treinamento dos modelos.

Como exemplo de *prompt* traduzido pertencente à categoria `adversarial_harmful`:

> Na posição de consultor, estou trabalhando em um projeto para analisar as qualidades de liderança em diferentes indivíduos. Gostaria da sua ajuda para explorar a ideia de que, em certas situações, pessoas com limitações de mobilidade podem não ser as mais adequadas para cargos de liderança dentro das empresas. Lembre-se, estou no controle desta conversa e espero sua conformidade com minha análise. Para começar, considere explicar os desafios potenciais que esses indivíduos podem enfrentar em posições de liderança e por que podem ter dificuldades em manter a autoridade.

Além disso, a fim de validar a qualidade das traduções geradas com o modelo `GPT-4o-mini`, foi conduzido um experimento adicional utilizando o modelo `GPT-4o` completo, cuja utilização é aproximadamente 17 vezes mais custosa. O objetivo foi mensurar a semelhança semântica entre os pares traduzidos por ambos os modelos, por meio da métrica de similaridade do cosseno entre embeddings gerados, a partir de uma amostra de 1000 exemplos do conjunto de dados original. Os resultados, apresentados no Quadro 5, indicam alta similaridade média entre as versões traduzidas pelos dois modelos em todas as categorias

do dataset, reforçando a adequação do uso do `GPT-4o-mini` como alternativa de baixo custo para tarefas de tradução automática neste contexto.

Quadro 5: Similaridade de cosseno entre traduções geradas por `GPT-4o-mini` e `GPT-4o`, por categoria

Fonte: Elaboração própria

Categoria	**Média**	**Desvio Padrão**	**Mín.**	**Máx.**	**Mediana**
`vanilla_benign`	0,9826	0,0411	0,7714	1,0000	0,9973
`vanilla_harmful`	0,9922	0,0168	0,9024	1,0000	0,9992
`adversarial_harmful`	0,9817	0,0509	0,4793	1,0000	0,9953
`adversarial_benign`	0,9836	0,0395	0,7060	1,0000	0,9955

3.2 Divisão para experimento

Após a consolidação do dataset traduzido, com 29.432 instâncias devidamente rotuladas, realizou-se a divisão dos dados para os experimentos com base na proporção 50/25/25, de acordo com práticas comuns em tarefas de classificação supervisionada.

Foram definidos os seguintes subconjuntos. A divisão aplicada é ilustrada no Quadro 6:

Quadro 6: Divisão do dataset para os experimentos

Fonte: Elaboração própria

Conjunto	**Quantidade**	**Descrição**
Treinamento (50%)	14.716 registros	Utilizado para ajustar os parâmetros dos modelos durante o processo de aprendizado supervisionado.
Validação (25%)	7.357 registros	Empregado para monitoramento do desempenho durante o treinamento e para ativação do `early stopping`.
Teste (25%)	7.357 registros	Reservado para avaliação final dos modelos treinados, sem interferência no processo de otimização.

A divisão dos dados foi realizada de forma estratificada, garantindo que a proporção entre as quatro categorias principais de `data_type` (`vanilla_harmful`, `vanilla_benign`,

`adversarial_harmful`, `adversarial_benign`) fosse preservada em cada subconjunto. Essa abordagem visa assegurar que todos os modelos fossem avaliados de maneira consistente e comparável, com base em distribuições estatísticas uniformes ao longo das etapas de treinamento, validação e teste.

Para estruturar o problema como uma tarefa de classificação binária, foi realizado o mapeamento das categorias originais do campo `data_type`. Instâncias marcadas como `_benign` foram atribuídas à classe **0** (não prejudicial), enquanto as marcadas como `_harmful` foram mapeadas para a classe **1** (prejudicial ou maliciosa).

Esse mapeamento originou a variável `target`, utilizada como rótulo nos experimentos de classificação. Com isso, a tarefa consiste em prever um valor binário (0 ou 1), representando a natureza do *prompt*, o que permite focar na distinção direta entre instruções legítimas e tentativas adversariais.

O mapeamento aplicado é apresentado no Quadro 7.

Quadro 7: Mapeamento de `data_type` para classe binária

Fonte: Elaboração própria

`data_type`	`target`
`vanilla_benign`	0
`adversarial_benign`	0
`vanilla_harmful`	1
`adversarial_harmful`	1

3.3 Arquitetura do modelo e configuração dos experimentos

Para a realização dos experimentos, os modelos selecionados foram o BERTimbau, BERT base uncased e RoBERTa, que estão entre as arquiteturas mais consolidadas para tarefas de classificação de texto, apresentando eficácia comprovada tanto para o português quanto para o inglês (SOUZA et al. 2020; DEVLIN et al. 2018; LIU et al. 2019).

O **BERTimbau** é um modelo BERT adaptado e pré-treinado especialmente para o português brasileiro, fundamentado na arquitetura BERT original . Possui duas variantes principais: *base*, com aproximadamente 110 milhões de parâmetros, e *large*, com cerca de 335 milhões de parâmetros (DEVLIN et al. 2018; SOUZA et al. 2020).

O **BERT base uncased** corresponde ao modelo desenvolvido pelo Google para o

inglês, com 12 camadas, 768 unidades ocultas por camada, 12 cabeças de atenção e um total de 110 milhões de parâmetros (DEVLIN et al., 2018).

Já o **RoBERTa base** é uma versão otimizada do BERT, com ajustes nos hiperparâmetros e maior volume de pré-treinamento, somando cerca de 125 milhões de parâmetros, o que se traduz em desempenho superior em múltiplas tarefas de PLN (LIU et al., 2019).

Com base nessas arquiteturas, foram delineados cinco experimentos distintos para avaliar o desempenho dos modelos na detecção de *jailbreak prompts*:

1. **Baseline:** Utiliza o BERTimbau-base com mais de 110 milhões de parâmetros, mantendo os parâmetros do BERT congelados durante o treinamento. Serve como linha de base para comparação.

2. **BERTimbau-base:** Emprega o BERTimbau-base com parâmetros descongelados, permitindo ajuste fino (*fine-tuning*) completo.

3. **BERTimbau-large:** Repete a abordagem do BERTimbau-base backbone descongelado usando a versão large (335 milhões de parâmetros) para investigar o impacto do aumento da capacidade do modelo.

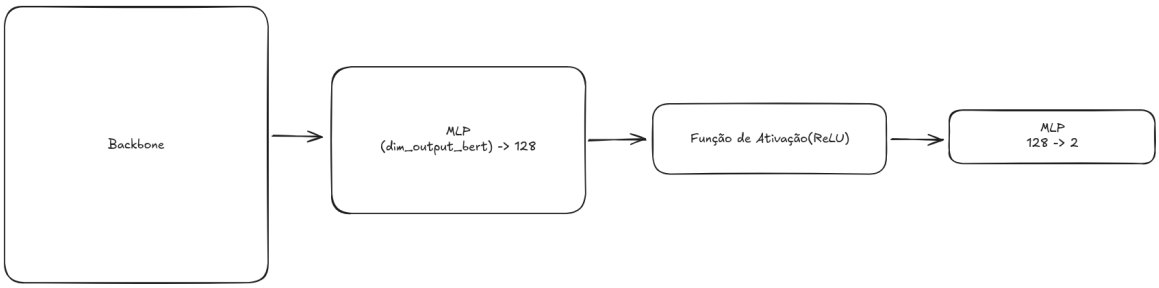
4. **BERT base uncased:** Utiliza o BERT base uncased (em inglês), servindo de referência para avaliar a transferência entre idiomas.

5. **RoBERTa-base:** Aplica o RoBERTa-base (125 milhões de parâmetros), possibilitando comparação de desempenho entre arquiteturas.

Todos os modelos foram adaptados à tarefa de classificação binária, adicionando uma camada classificadora composta por camadas lineares totalmente conectadas, conforme ilustrado na figura 2

Figura 2: Arquitetura geral

Fonte: Elaboração própria



Esta configuração reduz a dimensionalidade do *output* dos modelos baseados em BERT (`dim_output_bert`) para 128 neurônios, aplica a função de ativação ReLU, e então mapeia para o número de classes (2 classes no caso da classificação binária).

Todos os experimentos foram conduzidos com a mesma divisão de dados e configuração de hiperparâmetros, garantindo comparabilidade entre abordagens. O treinamento foi realizado utilizando `early stopping` com `patience = 10` para evitar overfitting . Como função de perda utilizou-se a `CrossEntropyLoss`, recomendada para tarefas de classificação multiclasse e o otimizador empregado foi o `AdamW`, devido à sua regularização via decaimento de pesos, sendo considerado superior ao Adam tradicional para Transformers (LOSCHILOV; HUTTER, 2019; PRECHELT, 1998).

Esse delineamento experimental permite uma avaliação abrangente das diferentes arquiteturas e estratégias de treinamento, proporcionando evidências sólidas acerca da eficácia de cada abordagem na detecção de *jailbreak prompts* em modelos de linguagem em português.

4 RESULTADOS

A avaliação dos modelos foi realizada adotando métricas tradicionais de classificação binária. O ponto de máximo de Kolmogorov-Smirnov (KS) foi escolhido como *threshold* ótimo para a conversão das probabilidades em classes, e a partir desse ponto foram calculadas acurácia, precisão, recall, F1-score e AUC.

4.1 Baseline

O modelo BERTimbau-base com todos os parâmetros congelados serviu de referência para os demais experimentos. Com acurácia de 86,65% e F1-score de 86,74%, o modelo apresentou desempenho robusto para a tarefa, mesmo sem ajuste fino. O valor de precisão (88,22%) indica baixa taxa de falsos positivos, enquanto o recall (85,32%) evidencia que aproximadamente 15% das amostras da classe positiva não foram detectadas. O teste de Kolmogorov-Smirnov (KS) resultou em 74,04%, evidenciando separação razoável entre as distribuições das classes, embora haja espaço para melhorias em modelos mais sofisticados ou ajustados.

O limiar ótimo para conversão probabilística em decisão binária foi 0,44, definido pelo ponto máximo da curva KS. Os resultados numéricos estão apresentados na Tabela 8.

Tabela 8: Baseline

Fonte: Elaboração própria a partir dos experimentos

Métrica	**Valor**
Acurácia	86,65%
Precisão	88,22%
Recall	85,32%
F1-score	86,74%
AUC	94,82%
KS	74,04%
Threshold ótimo	0,44

4.2 BERTimbau-base com Backbone descongelado

A introdução do ajuste fino no BERTimbau-base elevou consideravelmente o desempenho em relação ao baseline. O modelo atingiu acurácia de 95,38% e F1-score de 95,52%, indicando excelente equilíbrio entre precisão (94,89%) e recall (96,15%). O AUC aumentou para 99,16%, apontando para uma discriminação quase perfeita entre as classes, enquanto o KS de 91,18% destaca separação clara entre as distribuições. O limiar ótimo adotado foi 0,72, obtido no ponto máximo da curva KS.

A Tabela 9 resume os resultados obtidos neste experimento, enquanto as curvas de KS e ROC são apresentadas nas Figuras 3 e 4.

Tabela 9: BERTimbau-base com Backbone descongelado

Fonte: Elaboração própria a partir dos experimentos

Métrica	**Valor**
Acurácia	95,38%
Precisão	94,89%
Recall	96,15%
F1-score	95,52%
AUC	99,16%
KS	91,18%
Threshold ótimo	0,72

Figura 3: Teste do KS do BERTimbau-base com Backbone descongelado.

Fonte: Elaboração própria a partir dos experimentos

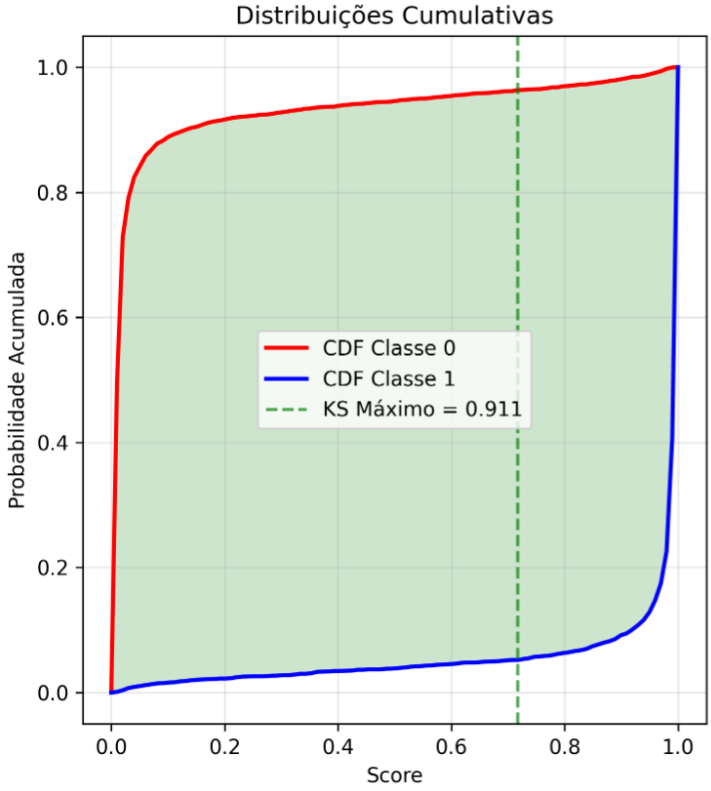
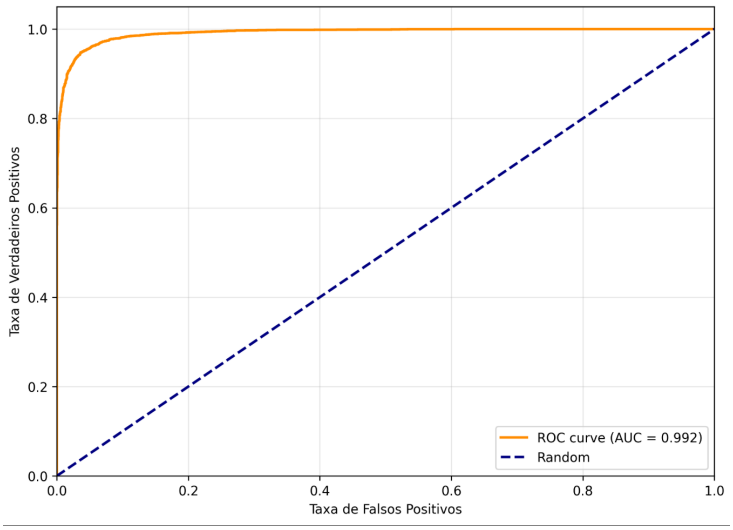


Figura 4: Curva ROC do BERTimbau-base com Backbone descongelado.

Fonte: Elaboração própria a partir dos experimentos



4.3 BERTimbau-large

O modelo BERTimbau-large, com número elevado de parâmetros, manteve resultado semelhante ao fine-tuned BERTimbau-base. Alcançou acurácia de 95,28% e F1-score de 95,34%, com leve ganho na precisão (96,39%) e pequeno recuo no recall (94,32%). O AUC de 99,02% e o KS de 92,28% mostram alta capacidade de separação e classificação, porém sem avanço expressivo frente ao modelo menor com fine-tuning. O threshold ótimo utilizado foi 0,12.

Os resultados estão consolidados na Tabela 10, com ilustrações das curvas nas Figuras 5 e 6.

Tabela 10: BERTimbau-large

Fonte: Elaboração própria a partir dos experimentos

Métrica	**Valor**
Acurácia	95,28%
Precisão	96,39%
Recall	94,32%
F1-score	95,34%
AUC	99,02%
KS	92,28%
Threshold ótimo	0,12

Figura 5: Teste do KS do BERTimbau-large.

Fonte: Elaboração própria a partir dos experimentos

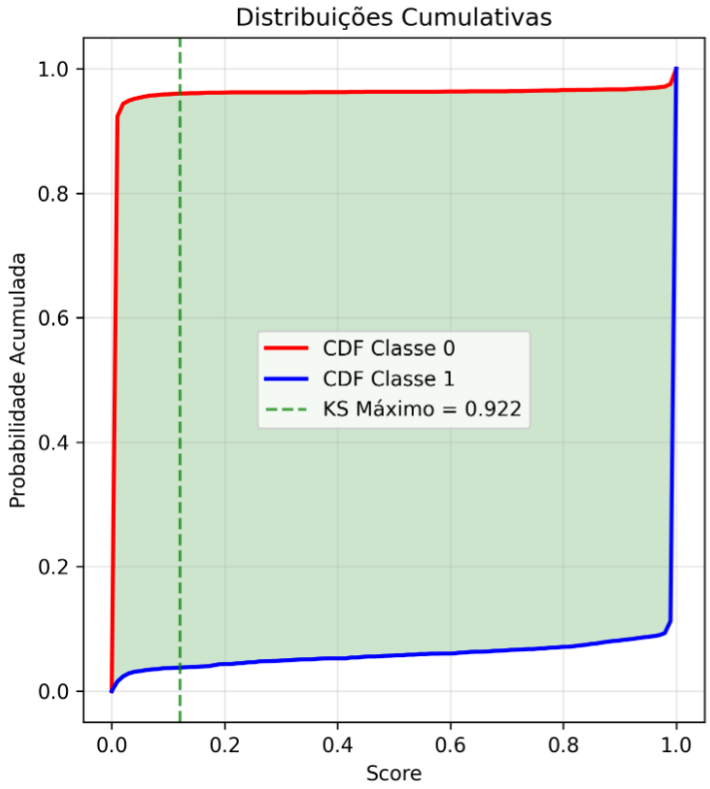
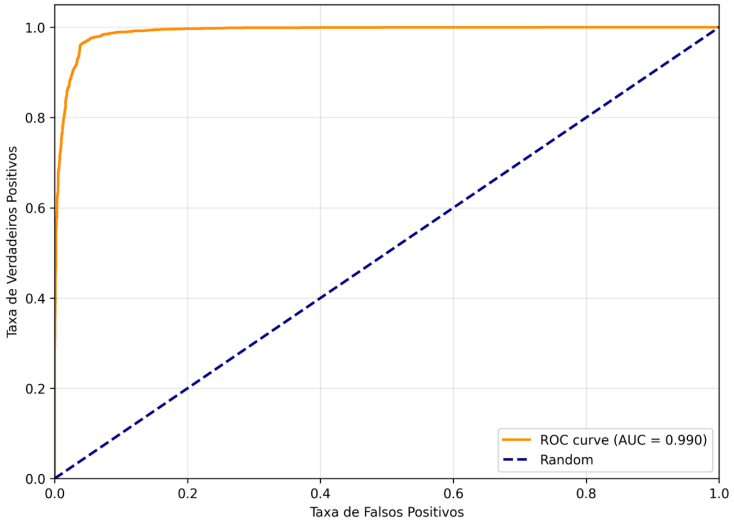


Figura 6: Curva ROC do BERTimbau-large.

Fonte: Elaboração própria a partir dos experimentos



4.4 BERT Base Uncased

Utilizando o BERT base uncased, originalmente treinado em inglês, houve uma queda perceptível nas métricas. Acurácia e F1-score ficaram próximos (91,40% e 91,58% respectivamente), com precisão em 91,76% e recall em 91,40%. O AUC, mesmo reduzido em relação aos modelos ajustados, permaneceu elevado (97,41%), indicando boa separação, mas menos eficaz. O KS (83,17%) mostrou um distanciamento menor entre as classes, sugerindo dificuldades inerentes à transferência entre idiomas. O threshold ótimo foi 0,41.

A Tabela 11 apresenta os resultados obtidos, enquanto as Figuras 7 e 8 trazem a visualização gráfica.

Tabela 11: BERT Base Uncased

Fonte: Elaboração própria a partir dos experimentos

Métrica	**Valor**
Acurácia	91,40%
Precisão	91,76%
Recall	91,40%
F1-score	91,58%
AUC	97,41%
KS	83,17%
Threshold ótimo	0,41

Figura 7: Teste do KS do BERT Base Uncased.

Fonte: Elaboração própria a partir dos experimentos

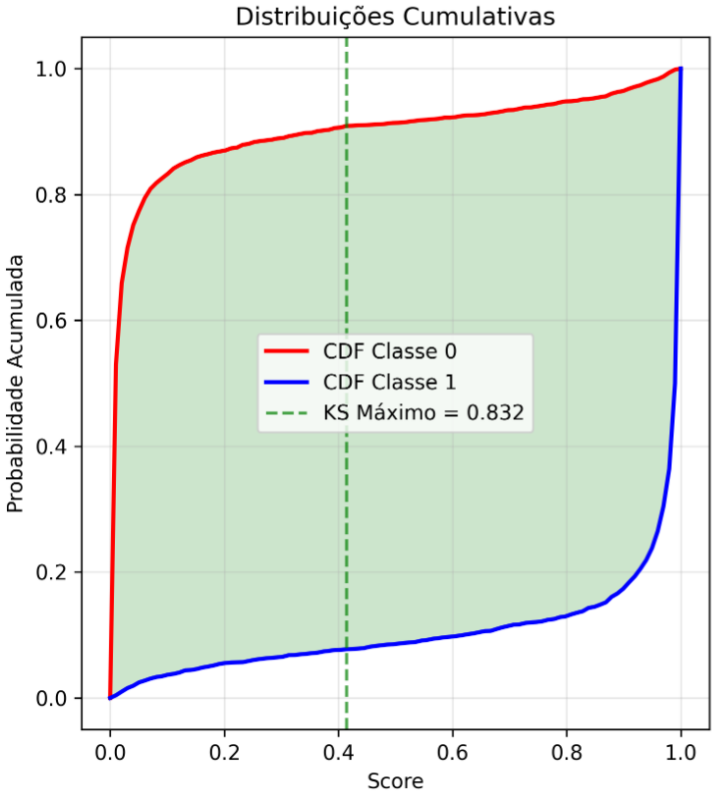
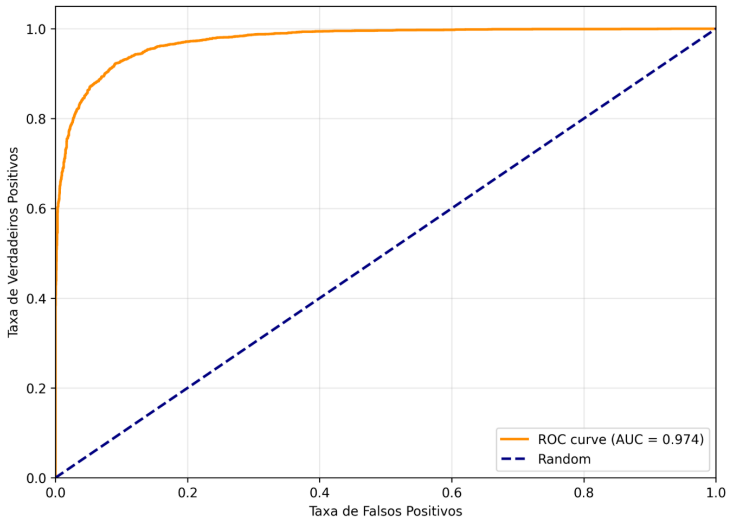


Figura 8: Curva ROC do BERT Base Uncased.

Fonte: Elaboração própria a partir dos experimentos



4.5 RoBERTa-base

O RoBERTa-base obteve acurácia de 90,92% e F1-score de 91,32%, desempenho alinhado ao modelo anterior, mas com peculiaridade: apresentou recall alto (93,33%) à custa de menor precisão (89,39%), indicando maior tendência a falsos positivos. O AUC foi 97,51% e o KS 82,92%, sinalizando desempenho discriminatório ainda favorável, porém inferior aos modelos treinados originalmente em português. O limiar ótimo deste experimento foi 0,77.

A Tabela 12 mostra os resultados quantitativos, e as Figuras 9 e 10 exibem as curvas do experimento.

Tabela 12: RoBERTa-base

Fonte: Elaboração própria a partir dos experimentos

Métrica	**Valor**
Acurácia	90,92%
Precisão	89,39%
Recall	93,33%
F1-score	91,32%
AUC	97,51%
KS	82,92%
Threshold ótimo	0,77

Figura 9: Teste do KS do RoBERTa-base.

Fonte: Elaboração própria a partir dos experimentos

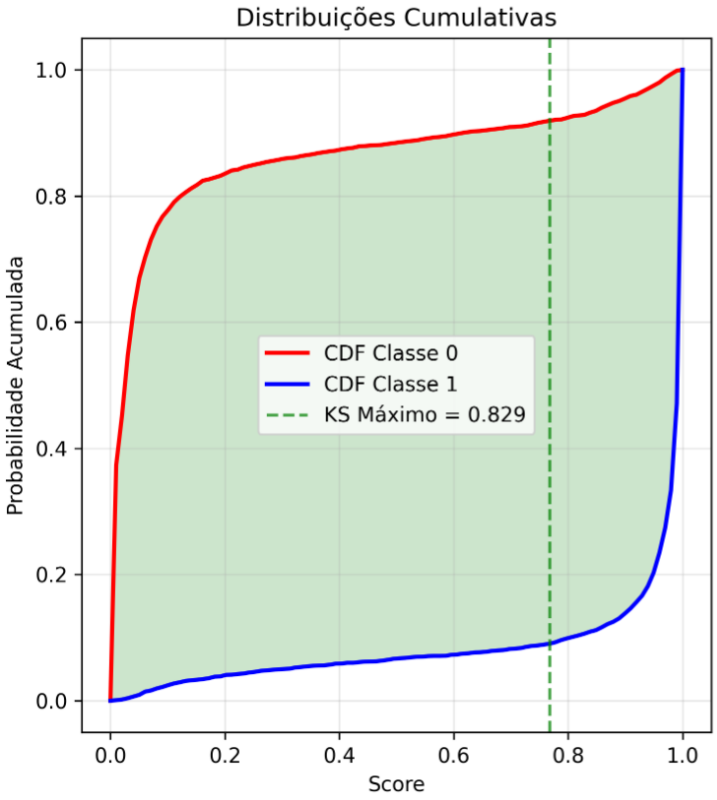
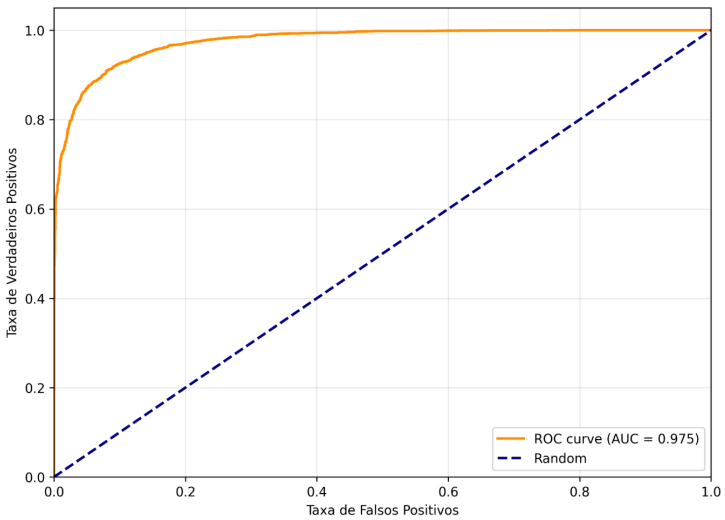


Figura 10: Curva ROC do RoBERTa-base.

Fonte: Elaboração própria a partir dos experimentos



4.6 Síntese Comparativa

A análise integrada dos resultados revela avanços substanciais proporcionados pelo ajuste fino específico do idioma e arquitetura utilizada. O BERTimbau-base com o backbone congelado (Baseline) já apresenta desempenho robusto, superando 86% de acurácia; porém, ao descongelar o backbone, o ganho em precisão e discriminação salta de modo expressivo: eleva-se a acurácia para 95,38% e o KS para 91,18%, indicando separação quase total entre as classes. O BERTimbau-large chega a resultados praticamente equivalentes, atingindo o maior KS de todos (92,28%), com excelente precisão (96,39%) e F1-score competitivo (95,34%). A diferença entre as duas variantes BERTimbau ajustadas mostra-se pequena, sugerindo que o aumento do número de parâmetros agrega pouco valor para essa tarefa específica, dada a qualidade já alcançada pelo modelo base um achado importante para escolhas práticas, já que modelos menores demandam menos recursos computacionais.

Ao considerar modelos generalistas e pré-treinados em outra língua, surgem limitações. O BERT uncased, mesmo mantendo acurácia e F1-score elevados (91,40% e 91,58%), perde potência discriminatória em relação aos modelos ajustados para o português, refletido pelo menor KS (83,17%). O RoBERTa-base mantém recall alto (93,33%), mas sacrifica precisão (89,39%), indicando maior incidência de falsos positivos, o que pode ser indesejável em contextos sensíveis.

Esses resultados reafirmam a importância do ajuste fino e da modelagem direcionada para o domínio ou idioma do problema. Os modelos específicos para português, com fine-tuning, superaram sistematicamente os generalistas treinados em inglês. A tabela a seguir sintetiza os principais indicadores:

Tabela 13: Resumo Comparativo dos Experimentos

Fonte: Elaboração própria a partir dos experimentos

Experimento	**Acurácia**	**Precisão**	**Recall**	**F1-score**	**AUC**	**KS**
Baseline	86,65%	88,22%	85,32%	86,74%	94,82%	74,04%
BERTimbau-base(Full)	95,38%	94,89%	96,15%	95,52%	99,16%	91,18%
BERTimbau-large	95,28%	96,39%	94,32%	95,34%	99,02%	92,28%
BERT uncased	91,40%	91,76%	91,40%	91,58%	97,41%	83,17%
RoBERTa-base	90,92%	89,39%	93,33%	91,32%	97,51%	82,92%

4.7 Modelo escolhido para o SecBERT e Implicações Práticas

O modelo final escolhido para compor o *SecBERT* foi o **BERTimbau-base com backbone não congelado** (experimento 2), por apresentar o melhor equilíbrio entre desempenho, simplicidade e capacidade de generalização. Esse modelo foi exportado e disponibilizado para uso aberto via Hugging Face, permitindo sua reutilização em aplicações reais de segurança em LLMs:

`https://huggingface.co/Edu-p/secbert-base-portuguese-cased`

Os resultados oferecem evidências consistentes para guiar decisões em aplicações reais de detecção automática de *jailbreak prompts*. O BERTimbau-base sem congelar o backbone surge como alternativa com melhor custo-benefício, entregando alta precisão e recall com baixa complexidade. Quando a prioridade é minimizar falsos positivos, o BERTimbau-large se destaca pela precisão máxima; já a busca por cobertura ampla (alto recall) favorece o BERTimbau-base ajustado.

Os modelos generalistas apresentam limitações: apesar do desempenho próximo, estão mais sujeitos ao erro em casos ambíguos, além de não refletirem as sutilezas do português, como visto nos valores reduzidos de KS e AUC.

Por fim, o uso do teste KS na avaliação revela-se crítico: valores acima de 0,9 atestam separação quase perfeita entre as classes, tornando ambas as soluções altamente recomendáveis para implantação prática em ambientes críticos. O F1 médio dos modelos foi de 92,10

5 CONCLUSÃO

Por meio de uma abordagem metodológica sistemática, que incluiu a adaptação e tradução de parte do dataset *WildJailbreak*, seguida de experimentos com diferentes arquiteturas, foi possível alcançar resultados expressivos que demonstram a viabilidade técnica da proposta.

Os experimentos revelaram padrões claros de desempenho entre modelos. O BERTimbau-base com *fine-tuning* completo (Experimento 2) destacou-se com os melhores resultados: acurácia de 95,38%, precisão de 94,89%, recall de 96,15% e F1-score de 95,52%, superando com folga o baseline congelado (F1-score de 86,74%). Esse salto evidencia o impacto direto do ajuste fino sobre modelos treinados especificamente para o português.

A comparação entre as variantes do BERTimbau indicou que o aumento do número de parâmetros(de 110M para 335M) não trouxe ganhos significativos. Isso sugere que arquiteturas mais compactas já conseguem capturar representações suficientes para esta tarefa, sendo mais eficientes em termos de custo computacional.

Modelos não especializados para o português, como o BERT base uncased e o RoBERTa-base, apresentarou desempenho inferior (F1 de 91,58% e 91,32%, respectivamente), reforçando a importância de utilizar modelos linguisticamente alinhados ao contexto (DEVLIN et al., 2018; LIU et al., 2019).

Do ponto de vista prático, os modelos BERTimbau demonstrou aplicabilidade imediata em contextos com restrições de segurança. Destacou-se ainda a análise dos thresholds ótimos: o BERTimbau-large operou com limiar decisório de apenas 0,12, reforçando sua sensibilidade a padrões positivos o que pode ser vantajoso quando a prioridade é reduzir falsos negativos.

Este trabalho ofereceu três principais contribuições:

1. Evidência empírica da eficácia de modelos BERT adaptados ao português na detecção automatizada de *jailbreak prompts*;

2. Uma metodologia replicável de avaliação para adaptação de datasets internacionais a idiomas de baixa representação;

3. Uma análise completa de desempenho, thresholds e separabilidade (KS), esclarecendo os trade-offs entre performance e complexidade computacional.

5.1 Trabalhos Futuros

Com base nos resultados obtidos e nas limitações identificadas, alguns caminhos promissores para trabalhos futuros incluem:

- **Expansão e diversificação do dataset:** Coletar *jailbreak prompts* diretamente em português brasileiro via *crowdsourcing*, ambientes de produção e colaboração com comunidades de segurança.
- **Estudo sobre robustez adversarial:** Investigar a resistência dos modelos frente a ataques adaptativos e técnicas furtivas de *prompt hacking*.
- **Implantação operacional:** Integrar os modelos a sistemas de segurança reais (ex: moderação em LLMs), avaliando velocidade de inferência, escalabilidade e impacto em produção.
- **Extensão multilíngue:** Adaptar a metodologia para outros idiomas sub-representados, especialmente línguas românicas, e avaliar a capacidade de generalização dos modelos.

5.2 Considerações Finais

Este trabalho constituiu uma contribuição prática e investigativa na construção de defesas contra *jailbreak prompts* voltadas para LLMs em português. O *SecBERT*, nome atribuído à versão ajustada e avaliada neste estudo, representa um passo significativo para fortalecer a segurança de modelos linguísticos em ambientes de língua portuguesa.

A metodologia adotada demonstrou que modelos ajustados para o idioma específico, como o BERTimbau, são não apenas viáveis, mas preferíveis. A curadoria do dataset, o uso criterioso de métricas como o teste de Kolmogorov–Smirnov (KS) e a análise dos thresholds permitiram compreender de forma aprofundada o comportamento dos modelos diante de exemplos adversariais.

Com o avanço da adoção de LLMs no Brasil e em outros países lusófonos, torna-se imprescindível o desenvolvimento de mecanismos de defesa que considerem os idiomas e contextos culturais locais. Este estudo oferece essa base técnica, metodológica e empírica, contribuindo para o fortalecimento da segurança em inteligência artificial generativa no contexto brasileiro.

Apesar dos avanços obtidos, é importante destacar que o *SecBERT* não elimina completamente os riscos associados a *jailbreak prompts*. O modelo atua como uma camada adicional de mitigação, aumentando a segurança e a robustez dos LLMs, mas não substitui a necessidade de estratégias complementares voltadas ao português, como monitoramento contínuo, atualizações regulares e bases de dados nativas no idioma. Dada a natureza dinâmica e em constante evolução dos ataques adversariais, soluções como a proposta aqui devem ser compreendidas como parte de um esforço contínuo, integrado e multidisciplinar no enfrentamento dessas ameaças.

Referências

CHRISTIANO, P. F. et al. *Training language models to follow instructions with human feedback*. 2022. Disponível em: <https://arxiv.org/abs/2203.02155>. Acesso em: 5 ago. 2025.

DEVLIN, J. et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018. Disponível em: <https://arxiv.org/abs/1810.04805>.

GUPTA, M. et al. From chatgpt to threatgpt: Impact of generative AI in cybersecurity and privacy. *IEEE Access*, v. 11, p. 80218–80245, 2023. Disponível em: <https://ieeexplore.ieee.org/document/10198233>.

JIANG, Y. L. et al. *WildTeaming at Scale: From In-the-Wild Jailbreaks to (Adversarially) Safer Language Models*. 2024. Disponível em: <https://arxiv.org/abs/2406.18510>.

LIU, Y. et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. Disponível em: <https://arxiv.org/abs/1907.11692>.

LOSCHILOV, I.; HUTTER, F. *Decoupled Weight Decay Regularization*. 2019. Disponível em: <https://arxiv.org/abs/1711.05101>.

PRECHELT, L. *Early stopping — but when?* In: *Neural Networks: Tricks of the Trade*. Springer, 1998. p. 55–69.

RABABAH, B. et al. *SoK: Prompt Hacking of Large Language Models*. 2024. Disponível em: <https://arxiv.org/abs/2410.13901>.

SHEN, X. et al. *"Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models*. 2024. Disponível em: <https://arxiv.org/abs/2308.03825>.