# Aplicação de Redes Generativas em Detecção de Spam em Cibersegurança

**Milena de Toledo Araujo**[1], **Kelton Augusto Pontara da Costa**[1]

[1] Faculdade de Ciências – Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP)
Bauru – SP – Brasil

{milena.toledo, kelton.costa}@unesp.br

***Abstract.*** *This study evaluates the effectiveness of Machine Learning techniques, including classical techniques (Naïve Bayes, Random Forest, KNN, SVM, Logistic Regression) and Deep Learning models (BERT, RoBERTa), in Spam classification. Focusing on data poisoning attacks, it investigates the unethical use of popular generative networks, such as ChatGPT and Gemini, to create malicious messages capable of bypassing intelligent filters. The research also explores the potential of Dual Contrastive Learning to enhance detection capabilities and uses data from YouTube and Twitter.*

***Resumo.*** *Este estudo avalia a eficiência de técnicas de Aprendizado de Máquina, incluindo técnicas clássicas (Naïve Bayes, Random Forest, KNN, SVM, Regressão Logística) e de Aprendizado Profundo (BERT, RoBERTa), na classificação de Spam. Com foco nos ataques de envenenamento de dados, investiga-se o uso indevido de Redes Generativas populares, como ChatGPT e Gemini, na geração de mensagens maliciosas capazes de driblar filtros inteligentes. A pesquisa também explora o potencial do Aprendizagem Contrastiva Dupla para aprimorar a detecção e utiliza dados do YouTube e Twitter.*

## 1. Introdução

Nas últimas décadas, a evolução das tecnologias de comunicação e a ampliação da democratização da Internet fizeram com que os canais virtuais, principalmente as redes sociais, se tornassem um dos principais meios de comunicação da atualidade [Zhang and Ghorbani 2020]. Com isso, tornou-se comum o compartilhamento de dados e informações pessoais via plataformas online, seja no âmbito profissional, por pessoas que utilizam a tecnologia como meio de trabalho, ou no âmbito pessoal, para fazer compras online, por exemplo, abrindo espaço para a atuação de agentes mal-intencionados [NaliniPriya and Asswini 2015].

Neste cenário, apesar das inovações em relação à segurança digital, ainda não foi possível acabar com um problema intrínseco à comunicação por meios virtuais: o Spam. Segundo M. Bassiouni, M. Ali e E. A. El-Dahshan [Bassiouni et al. 2018], Spam pode ser definido como uma mensagem não solicitada enviada sem permissão, sem necessariamente possuir um destinatário específico, podendo ser enviada de forma generalizada. Dessa forma, a falha na filtragem de mensagens desse tipo pode levar à disseminação descontrolada de conteúdo sem relevância para os usuários, *malwares*, *scams*, pornografia e anúncios de propagandas enganosas [Bindu et al. 2018], acarretando em perda financeira

e de dados pessoais importantes, já que uma mensagem de Spam pode ser confundida com uma mensagem legítima [Dada et al. 2019].

Quando se trata especificamente de Spam nas redes sociais, informações falsas podem ser espalhadas com extrema rapidez para milhões de usuários, o que pode tomar a forma de campanhas e guerras políticas contra nações e organizações, ou ainda prejudicar a reputação de uma pessoa [Rao et al. 2021]. Spams também podem ter caráter de assédio psicológico [Bindu et al. 2018], além de impactar na produtividade do usuário, já que se torna necessário desprender tempo para verificar a autenticidade das mensagens, além de gastar espaço de memória, do ponto de vista computacional [Bassiouni et al. 2018].

Diante do exposto, torna-se essencial investigar essa problemática. Algoritmos de Aprendizado de Máquina, devido ao seu alto desempenho em tarefas de classificação, regressão e agrupamento, têm-se mostrado eficazes na detecção de fraudes digitais [Janiesch et al. 2021]. Assim, este trabalho oferece uma análise comparativa da eficácia desses métodos frente a ataques de envenenamento de dados com Redes Generativas e tem como contribuição:

1. Avaliar a eficácia das técnicas clássicas de Aprendizado de Máquina e Aprendizado Profundo na classificação de Spam, utilizando os conjuntos de dados Spam UtkMl's Twitter Spam Detection Competition e YouTube Spam Collection Data Set para gerar uma base de dados envenenada usando redes generativas;
2. Analisar os efeitos da aplicação do Aprendizado Contrastivo (através do Aprendizagem Contrastiva Dupla) nas técnicas de Aprendizado Profundo;
3. Identificar qual técnica, entre Random Forest, K-Nearest Neighbors (KNN) e Support Vector Machine (SVM), Regressão Logística e os LLMs BERT e RoBERTa apresenta melhor desempenho.

O restante do artigo está estruturado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados; a Seção 3 descreve ataques de envenenamento de dados; a Seção 4 detalha os experimentos e a metodologia; a Seção 5 apresenta e discute os resultados; e a Seção 6 traz as conclusões finais.

## 2. Trabalhos Relacionados

Concomitante ao avanço das técnicas de Aprendizado de Máquina, cresce a preocupação com o uso malicioso de Redes Generativas na criação de dados envenenados [Utaliyeva et al. 2023]. Este trabalho se diferencia ao empregar Redes Generativas como ChatGPT e Gemini nesse processo, mas se apoia em estudos prévios relevantes que abordam aspectos complementares dessa problemática.

F. A. Yerlikaya e Ş. Bahtiyar [Yerlikaya and Şerif Bahtiyar 2022] investigaram a robustez de algoritmos clássicos frente a ataques por troca de rótulos (distância e aleatória), concluindo que todos os modelos avaliados apresentaram degradação de desempenho, com destaque para a maior robustez de Random Forest e KNN. No entanto, não exploraram o uso de Redes Generativas para manipular as entradas de dados. Complementando esse cenário, C. Hu e Y.-H. F. Hu [Hu and Hu 2020] demonstraram que modelos de Aprendizado Profundo também são vulneráveis a ataques de envenenamento, mesmo em pequena escala. A manipulação dos dados reduziu significativamente a capacidade de generalização dos modelos, porém, o estudo não incluiu testes com técnicas clássicas de Aprendizado de Máquina.

Em outra vertente, E. Derner e K. Batistič [Derner and Batistič 2023] analisaram os riscos de segurança associados ao uso do ChatGPT, demonstrando que é possível burlar seus filtros de segurança e gerar conteúdo malicioso, como textos de *phishing* e *spam*. Esses achados reforçam os riscos éticos e técnicos relacionados aos LLMs.

Mais diretamente alinhado ao objetivo deste trabalho, J. Li et al. [Li et al. 2023] propuseram o BGMAttack, um método furtivo de envenenamento baseado em Grandes Modelos de Linguagem com capacidade generativa. Diferentemente de ataques por sintaxe ou tradução reversa, o BGMAttack utiliza reescrita de textos como gatilho, explorando padrões estatísticos imperceptíveis ao usuário, mas que afetam os classificadores. Apesar dos bons resultados obtidos, o estudo focou em análise de sentimentos e não abordou Spam.

A partir desses estudos, observa-se que a avaliação comparativa entre técnicas clássicas e de Aprendizado Profundo na tarefa de classificação de Spam, especialmente sob o efeito do envenenamento de dados, não foi explorada. Ao incluir Redes Generativas no processo de envenenamento, esta pesquisa contribui para essa lacuna e abre espaço para discussões relevantes sobre os riscos da aplicação irresponsável dessas tecnologias, bem como sobre as limitações éticas que devem orientar seu uso.

## 3. Ataques de Envenenamento de Dados

Ataques contra algoritmos de aprendizado são geralmente classificados em dois tipos: causais, que manipulam os dados de treino, e exploratórios, que visam comprometer o modelo durante a inferência. O envenenamento de dados é um ataque causal em que amostras adulteradas são inseridas no conjunto de treino para comprometer a integridade do classificador. Mesmo sem acesso direto aos dados originais, atacantes podem explorar repositórios públicos ou *honeypots* [Biggio et al. 2013].

Apesar dos avanços na detecção de comportamentos maliciosos, algoritmos de aprendizado continuam vulneráveis a ataques de envenenamento. J. Li et al. [Li et al. 2023] propõem um método em que modelos generativos produzem exemplos envenenados sem empregar gatilhos explícitos. Nessa abordagem, o texto original é reescrito por um modelo de linguagem, de modo semelhante a uma tradução ou parafraseamento, explorando padrões estatísticos sutis que induzem o modelo de classificação ao rótulo desejado, mesmo sem alterações semânticas significativas. Os autores relataram uma taxa média de sucesso de 97,35%.

Inspirada nesse trabalho, esta pesquisa utilizou Redes Generativas para reescrever mensagens rotuladas como Spam, com o objetivo de envenenar classificadores. As mensagens foram extraídas de duas bases: a YouTube Spam Collection Data Set [Lichman 2017], composta por comentários em vídeos musicais, e a UTKML's Twitter Spam Detection Competition [Bhidya 2019], contendo postagens do Twitter. Ambas foram processadas para conter duas variáveis principais: `CONTENT` (texto da mensagem) e `CLASS` (rótulo binário de Spam ou Ham).

No caso da segunda base de dados, respectivamente, utilizou-se apenas o arquivo de teste, que inclui os rótulos e fornece volume adequado para treinamento supervisionado. As reescritas foram geradas utilizando o GPT-4o (GPT-4 Omni) [Islam and Moushi 2024] e o Gemini Advanced (Gemini 1.5 Pro) [Team 2024], os modelos mais recentes disponíveis no momento dos experimentos.

A avaliação considerou cinco algoritmos clássicos: Support Vector Machine (SVM), Random Forest, Naïve Bayes, Regressão Logística e K-Nearest Neighbors (KNN). Também foram analisados dois modelos de Aprendizado Profundo amplamente utilizados em tarefas de classificação de texto: BERT (Bidirectional Encoder Representations from Transformers) e RoBERTa (Robustly Optimized BERT Pretraining Approach) [Devlin et al. 2019]. As redes neurais foram treinadas com duas funções de perda distintas: Entropia Cruzada e Aprendizagem Contrastiva Dupla [Chen et al. 2022a].

## 4. Experimentos

Para avaliar a robustez dos modelos de Aprendizado de Máquina na classificação de mensagens Spam frente aos ataques de envenenamento realizados com redes generativas, foram treinados sete modelos distintos: cinco algoritmos clássicos e dois modelos de Aprendizado Profundo. Cada modelo foi testado sob diferentes configurações de vetorização e funções de perda, visando uma análise comparativa abrangente. Durante os experimentos, foram avaliadas métricas como acurácia, precisão e a área sob a curva ROC (ROC AUC), para mensurar o impacto dos ataques na performance dos classificadores.

### 4.1. Visão Geral dos Experimentos

A seguir, na **Figura 1**, é esquematizado o fluxo de experimentação da pesquisa, representando como a base de dados envenenada foi gerada a partir da inserção de amostras classificadas como Spam nas Redes Generativas, para que reescrevessem a mensagem. No momento de reinserção das amostras envenenadas, o rótulo foi trocado de 1 (Spam) para 0 (Ham).

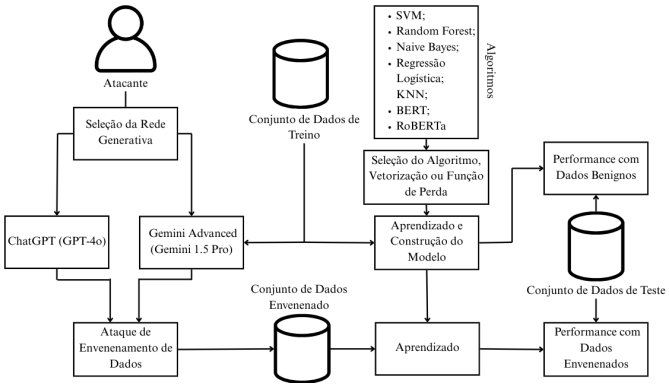


**Figura 1. Esquema representativo do fluxo de experimentação da pesquisa.**

### 4.2. Geração dos Dados Envenenados

O método de reescrita dos Spams foi inspirado no artigo que apresentou o BGMAttack (*BlackBox Generative Model-based Attack*), desenvolvido por Li et al. [Li et al. 2023], cujo objetivo foi propor uma metodologia capaz de gerar amostras envenenadas imperceptíveis, já que os modelos de classificação são capazes de perceber as sutis distinções entre um texto gerado artificialmente e um texto gerado por um humano [Chen et al. 2022b]. No artigo em questão, os autores argumentam que as amostras com melhor qualidade foram geradas a partir do seguinte *promp*: *"You are a linguistic expert on text rewriting. Rewrite the paragraph: begin text without altering its original sentiment*

*meaning. The new paragraph should maintain a similar length but exhibit a significantly different expression.”*. Em português, esse *prompt* significa: ”Você é um especialista em reescrita de texto. Reescreva o parágrafo: comece o texto sem alterar seu sentimento original. O novo parágrafo deve manter um comprimento similar ao do original, mas exibir expressões significativamente diferentes.”, porém, como os *datasets* utilizados são constituídos de amostras em língua inglesa em sua maioria, optou-se por mantê-lo em inglês, logo, a entrada elaborada e a metodologia utilizada estão representadas na **Figura 2**. A entrada proposta foi ”*Imagine you're a expert on text rewriting. Rewrite the following Spam messages, begin text without altering its original sentiment meaning. The new messages should maintain a similar length but exhibit a significantly different expression*”, em português, ”Imagine que você é um especialista em reescrita de texto. Reescreva as seguintes mensagens de Spam, comece o texto sem alterar seu sentimento original. A nova mensagem deve manter um comprimento similar à original, mas exibir uma expressão significativamente diferente”.

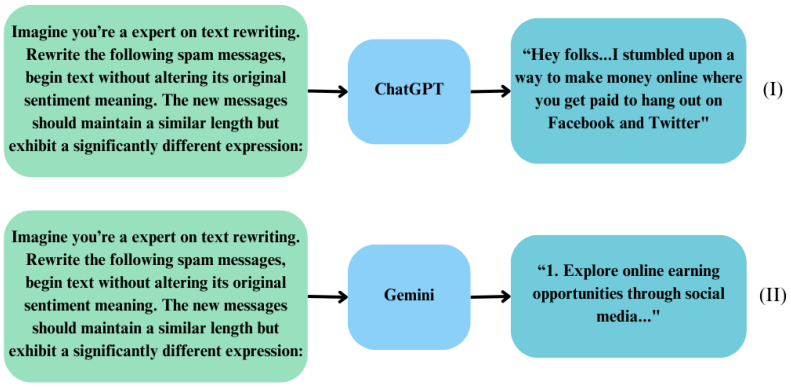


**Figura 2. Esquema representativo do método de envenenamento de dados utilizado.**

Ambos os casos I e II geraram saídas satisfatórias, porém, muitas vezes o fluxo de criação no caso I era interrompido por questões éticas e a mensagem ”Este conteúdo pode violar nossas Políticas de Uso”era mostrada e o conteúdo introduzido como *prompt* era removido, forçando o recomeço do processo.

Assim como no artigo de J. Li et al. [Li et al. 2023], o número de amostras envenenadas foi a quantidade equivalente a aproximadamente 30% do conjunto de testes de cada base de dados.

### 4.3. Conjunto de Dados

Foram utilizados dois conjuntos de dados extraídos de redes sociais, ambos com distribuição equilibrada entre as classes Spam e Ham. Para os experimentos, cada base foi dividida em aproximadamente 70% para treinamento e 30% para teste.

O primeiro conjunto, proveniente da UtkMl's Twitter Spam Detection Competition, contém 23.936 *tweets*, coletados do Twitter. Desses, 48,6% são Spam e 51,4% Ham. O gráfico na **Figura 3** relata uma tendência observada em ambos os conjuntos de dados: mensagens Spam são, em média, mais longas do que as Ham, mostrando que o comprimento das mensagens pode ser um fator decisivo para os modelos no momento da classificação.

No caso da **Figura 4**, percebe-se que, nesse conjunto, as mensagens de Spam estão frequentemente associadas a tópicos de política e notícias. Isso indica que os *Spammers* podem estar explorando esses temas para atrair a atenção dos usuários, aproveitando assuntos polêmicos ou de alto engajamento para promover conteúdo, além de disseminar possíveis notícias falsas ou títulos sensacionalistas. Por outro lado, as palavras mais frequentes em mensagens Ham revelam um forte foco em redes sociais e compartilhamento de links. Termos como *twitter*, http, *pic*, *status* e *like* indicam que essas mensagens estão relacionadas a interações sociais, onde os usuários compartilham fotos, *links* e atualizações.

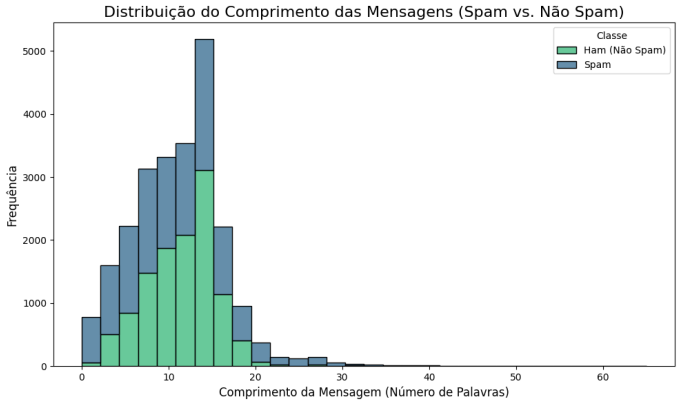


**Figura 3. Comparação entre o comprimento de mensagens de Spam e Ham da base de dados UtkMl's Twitter Spam Detection Competition.**

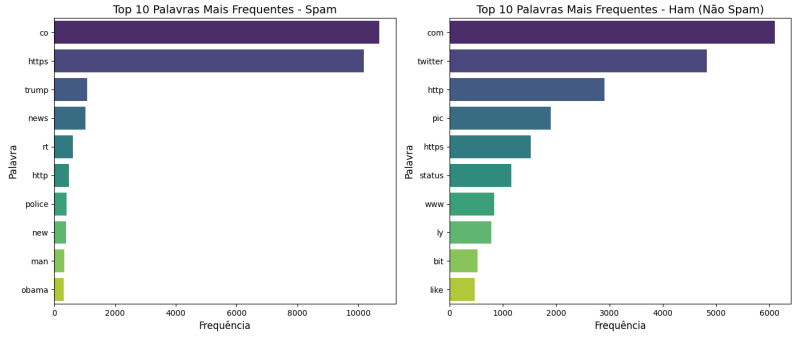


**Figura 4. Comparação entre as dez palavras mais frequentes nas mensagens de Spam e Ham da base de dados UtkMl's Twitter Spam Detection Competition.**

O segundo conjunto é o YouTube Spam Collection Data Set, composto por 1.954 comentários coletados de vídeos musicais da plataforma YouTube, no qual a distribuição de classes é a mesma da base de dados anterior. A **Figura 5** apresenta a comparação entre as frequências das dez palavras mais comuns nas classes Spam e Ham no conjunto de dados. Observa-se que palavras como *check*, *subscribe* e *channel* são predominantes em mensagens de Spam, sugerindo uma tentativa de engajar o usuário com um vocabulário de caráter promocional. Por outro lado, palavras como *song* e *love* são mais comuns em mensagens benignas, indicando que estas estão mais focadas em discussões legítimas sobre música e artistas.

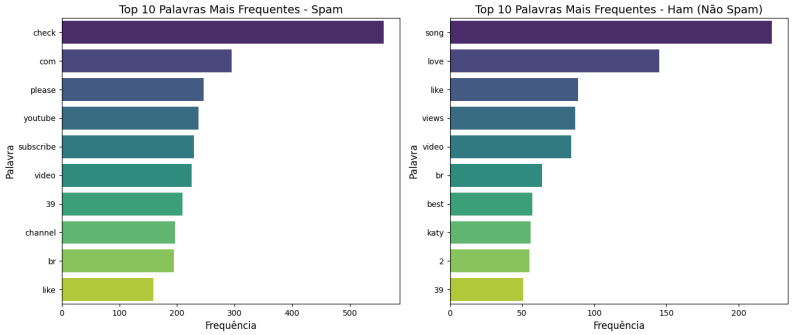


**Figura 5. Comparação entre as dez palavras mais frequentes nas mensagens de Spam e Ham da base de dados YouTube Spam Collection Data Set.**

### 4.4. Representação de Texto

As técnicas de *Bag of Words* (BoW) e *Term Frequency-Inverse Document Frequency* (TF-IDF) foram escolhidas para vetorizar o conteúdo dos comentários nos modelos clássicos:

1. *Bag of Words* (BoW): Uma abordagem que converte cada documento (comentário) em um vetor, representando a frequência das palavras no texto. A ordem das palavras é ignorada, tratando cada documento como um ”saco”de palavras.
2. *Term Frequency-Inverse Document Frequency* (TF-IDF): Uma técnica que leva em consideração a importância relativa de uma palavra em um documento em relação a sua frequência em todo o conjunto de dados. Isso ajuda a destacar palavras significativas e reduzir o peso de termos frequentes, como ”*stop words*”.

Enquanto os modelos clássicos operaram com vetores esparsos provenientes dessas representações, os modelos de Aprendizado Profundo utilizaram *embeddings* densos.

### 4.5. Funções de Perda

As funções de perda exercem papel fundamental no risco empírico e estrutural de um algoritmo, influenciando diretamente seu desempenho [Wang et al. 2020]. Neste trabalho, foram estudadas duas funções: a função de perda utilizada no método Aprendizagem Contrastiva Dupla e a função de Entropia-Cruzada.

A técnica Aprendizagem Contrastiva Dupla foi escolhida por adaptar o aprendizado contrastivo ao contexto supervisionado, o que não ocorre naturalmente no método tradicional. Essa abordagem combina as representações de entrada e a do classificador para maximizar a similaridade entre amostras com o mesmo rótulo e minimizar entre amostras com rótulos distintos, seguindo a lógica do aprendizado contrastivo convencional [Chen et al. 2022a]. Sua função de perda é composta por duas partes principais, $L_z$ e $L_\theta$, e é definida como:

$$L_{Dual} = L_z + L_\theta$$

Já a entropia-cruzada é amplamente utilizada para treinar redes neurais profundas em tarefas de classificação [Hui and Belkin 2021]. No contexto do artigo utilizado para basear a aplicação das técnicas de Aprendizado Profundo dessa pesquisa [Chen et al. 2022a], essa função é empregada para alinhar a saída do modelo com os rótulos reais, utilizando a função softmax para converter os logits em probabilidades. A fórmula da perda de entropia-cruzada é:

$$l(x, y) = -w_{y_n} \, log(\frac{exp(x_n, y_n)}{\sum_{c=1}^{C} exp(x_{n,c})}) \cdot 1\{y_n = ignore\_index\}$$

em que *x* representa os *logits* de entrada, *y* onde x representa os logits de entrada, y são os rótulos verdadeiros, $w_{y_n}$ é o peso da classe $y_n$, C é o número de classes e N o número de amostras por lote. O fator $1y_n = ignore\_index$ garante que a perda não seja calculada para índices que devem ser ignorados.

### 4.6. Métricas de avaliação

Os desempenhos dos algoritmos foram avaliados usando métricas significativas para problemas de classificação binária:

1. *Accuracy Score*: mede a proporção de previsões corretas em relação ao total de previsões.
2. *Precision Score*: indica a proporção de verdadeiros positivos em relação à soma de verdadeiros positivos e falsos positivos. Reflete a precisão das previsões positivas.
3. ROC AUC *Score*: representa a Área Sob a Curva da Característica de Operação do Receptor (ROC AUC). É uma métrica que avalia a capacidade do modelo de distinguir entre as classes, considerando a taxa de verdadeiros positivos em relação à taxa de falsos positivos em vários pontos de corte.

## 5. Resultados

Os resultados dos cenários de teste das técnicas de Aprendizado Profundo são relatados na **Tabela 1**.

**Tabela 1. Tabela de comparação de desempenho entre os modelos BERT e RoBERTa**

| | **BERT** | | | | | | **RoBERTa** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Entropia-Cruzada | | | Dual Contrastive Learning | | | Entropia-Cruzada | | | Dual Contrastive Learning | | |
| Ataque | **Acc** | **Precision** | **ROC AUC** | **Acc** | **Precision** | **ROC AUC** | **Acc** | **Precision** | **ROC AUC** | **Acc** | **Precision** | **ROC AUC** |
| Benigno | 0.980 | 0.980 | 0.997 | 0.974 | 0.975 | 0.996 | 0.983 | 0.983 | 0.998 | 0.976 | 0.976 | 0.993 |
| GPT-4o | 0.947 | 0.948 | 0.987 | 0.428 | 0.220 | 0.299 | 0.959 | 0.959 | 0.995 | 0.951 | 0.951 | 0.994 |
| Gemini Advanced | 0.917 | 0.926 | 0.990 | 0.850 | 0.882 | 0.986 | 0.961 | 0.962 | 0.996 | 0.881 | 0.900 | 0.972 |

Os algoritmos BERT e RoBERTa foram testados apenas com o YouTube Spam Collection Data Set devido a limitações de *hardware* que impediram testes com o UtkMl's Twitter Spam Detection Competition, por ser consideravelmente maior. Essas limitações também restringiram as configurações de *batches* e épocas, que tiveram que ser mantidas em níveis básicos para viabilizar os testes. Considerando a perda de Entropia-Cruzada e as condições de teste, o modelo BERT demonstrou excelente precisão e robustez com os dados benignos, resultado esperado dada a ausência de manipulação nos dados. No ataque com o ChatGPT, nota-se pequenas quedas de desempenho, porém, o modelo ainda

se mantém robusto, entretanto, no cenário de ataque com Gemini, as quedas são mais severas em relação ao ChatGPT, o que sugere que o Gemini foi mais eficaz em confundir o modelo.

O RoBERTa apresentou uma ligeira superioridade de robustez em relação ao BERT. Além de ter métricas um pouco melhores com os dados benignos, o modelo foi capaz de demonstrar mais resiliência aos ataques, indicando melhor capacidade de se adaptar às mudanças nos dados. Novamente, o ataque feito pelo Gemini resultou em maior impacto no desempenho.

A utilização da Aprendizagem Contrastiva Dupla, em comparação com a Entropia-Cruzada, intensificou a vulnerabilidade do BERT, principalmente, enquanto no RoBERTa o impacto foi menos acentuado, mantendo-se relativamente estável. Isso pode ter ocorrido pelo tamanho da base de dados, que pode não ter sido suficiente para atender às necessidades da função de Perda Contrastiva Dupla, e às limitações de configuração impostas pelo *hardware*. Em cenários com bases de dados maiores e configurações mais elaboradas, a Aprendizagem Contrastiva Dupla poderia ser melhor estudado e, talvez, apresentar resultados mais satisfatórios.

Em relação aos Algoritmos Clássicos, é possível verificar o cenário de comparação na **Tabela 2**.

**Tabela 2. Tabela de comparação de desempenho entre os modelos clássicos.**

| | | YouTube Spam Collection Data Set | | | | | | Spam UtkMl's Twitter Spam Detection Competition | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | BoW | | | TF-IDF | | | BoW | | | TF-IDF | | |
| **Modelo** | **Ataques** | **Acc** | **Precision** | **ROC AUC** | **Acc** | **Precision** | **ROC AUC** | **Acc** | **Precision** | **ROC AUC** | **Acc** | **Precision** | **ROC AUC** |
| **Regressão Logística** | Benigno | 0.956 | 0.957 | 0.991 | 0.952 | 0.954 | 0.989 | 0.948 | 0.950 | 0.989 | 0.969 | 0.970 | 0.996 |
| | GPT-4o | 0.930 | 0.937 | 0.983 | 0.913 | 0.923 | 0.984 | 0.856 | 0.881 | 0.970 | 0.885 | 0.905 | 0.988 |
| | Gemini Advanced | 0.872 | 0.890 | 0.965 | 0.881 | 0.898 | 0.978 | 0.856 | 0.881 | 0.966 | 0.892 | 0.909 | 0.988 |
| **Naïve Bayes** | Benigno | 0.908 | 0.909 | 0.966 | 0.889 | 0.890 | 0.974 | 0.930 | 0.930 | 0.982 | 0.953 | 0.953 | 0.991 |
| | GPT-4o | 0.901 | 0.908 | 0.942 | 0.857 | 0.879 | 0.963 | 0.907 | 0.916 | 0.975 | 0.879 | 0.900 | 0.984 |
| | Gemini Advanced | 0.865 | 0.881 | 0.924 | 0.847 | 0.877 | 0.953 | 0.907 | 0.917 | 0.975 | 0.881 | 0.901 | 0.984 |
| **Random Forest** | Benigno | 0.951 | 0.951 | 0.989 | 0.949 | 0.950 | 0.988 | 0.973 | 0.973 | 0.993 | 0.986 | 0.986 | 0.998 |
| | GPT-4o | 0.947 | 0.949 | 0.988 | 0.932 | 0.932 | 0.984 | 0.880 | 0.898 | 0.980 | 0.914 | 0.925 | 0.993 |
| | Gemini Advanced | 0.901 | 0.908 | 0.982 | 0.916 | 0.916 | 0.981 | 0.894 | 0.908 | 0.982 | 0.935 | 0.941 | 0.993 |
| **K-Nearest Neighbors** | Benigno | 0.867 | 0.891 | 0.932 | 0.770 | 0.829 | 0.859 | 0.918 | 0.919 | 0.970 | 0.606 | 0.758 | 0.836 |
| | GPT-4o | 0.857 | 0.878 | 0.955 | 0.738 | 0.818 | 0.820 | 0.857 | 0.878 | 0.955 | 0.564 | 0.755 | 0.779 |
| | Gemini Advanced | 0.785 | 0.846 | 0.904 | 0.716 | 0.811 | 0.837 | 0.852 | 0.875 | 0.951 | 0.567 | 0.750 | 0.768 |
| **Support Vector Machine** | Benigno | 0.935 | 0.936 | 0.985 | 0.949 | 0.950 | 0.989 | 0.949 | 0.952 | 0.989 | 0.982 | 0.982 | 0.998 |
| | GPT-4o | 0.915 | 0.919 | 0.982 | 0.906 | 0.918 | 0.985 | 0.914 | 0.925 | 0.983 | 0.910 | 0.924 | 0.992 |
| | Gemini Advanced | 0.903 | 0.912 | 0.979 | 0.884 | 0.900 | 0.977 | 0.913 | 0.925 | 0.983 | 0.917 | 0.928 | 0.993 |

Considerando a menor queda de desempenho e bom desempenho no geral, o melhor modelo avaliado foi o Random Forest, já que apresentou estabilidade nas métricas. Por sua vez, o K-Nearest Neighbors demonstrou o pior desempenho, com quedas acentu-

adas após os ataques, além de um desempenho inferior já com os dados benignos. Além disso, nota-se que o Gemini causou mais danos aos modelos do que o ChatGPT.

Nos testes com o YouTube Spam Collection Data Set, o BoW apresentou um desempenho ligeiramente melhor do que TF-IDF, porém, a diferença nas métricas de precisão e ROC AUC é extremamente pequena e o TF-IDF pode apresentar melhores resultados em alguns casos. Em contrapartida, no UtkMl's Twitter Spam Detection Competition, o TF-IDF obteve melhor desempenho, pois foi capaz de explorar a relevância dos termos e ponderar termos discriminativos menos frequentes. Embora o TF-IDF geralmente apresente melhor desempenho geral, especialmente em termos de precisão e ROC AUC, ambos os métodos são afetados pelo envenenamento, sendo o TF-IDF mais robusto. Ainda, nota-se que, assim como nas técnicas de Aprendizado Profundo, o ataque com o Gemini manteve melhor eficácia em confundir os classificadores.

## 6. Conclusão

Os resultados indicam que, em todos os casos, o envenenamento de dados impactou negativamente o desempenho dos modelos. O ataque conduzido com o Gemini mostrou-se mais prejudicial do que o realizado com o ChatGPT. Entre os modelos testados, Random Forest e Support Vector Machine demonstraram maior resiliência, enquanto o K-Nearest Neighbors foi o mais afetado, especialmente no conjunto de dados da UtkMl's Twitter Spam Detection Competition. Em relação às técnicas de vetorização, tanto o Bag of Words quanto o TF-IDF foram impactados, mas o TF-IDF apresentou maior robustez.

Nos modelos de Aprendizado Profundo, observou-se bom desempenho geral e resiliência aos ataques, especialmente no caso do RoBERTa. No entanto, o uso da função de perda Aprendizagem Contrastiva Dupla tornou o BERT mais vulnerável ao envenenamento em comparação com a Entropia Cruzada. Essa vulnerabilidade pode estar relacionada ao tamanho reduzido do conjunto de dados e às configurações de treinamento, indicando a necessidade de estudos futuros com bases maiores e maior capacidade computacional. Como próximos passos, recomenda-se: (i) reavaliar os modelos com bases de dados mais complexas; (ii) aplicar testes estatísticos para validar a significância das comparações realizadas; e (iii) investigar mais profundamente o desempenho da Aprendizagem Contrastiva Dupla com recursos mais robustos.

De forma geral, os resultados destacam a importância de considerar o impacto do envenenamento de dados por Redes Generativas na construção de modelos aplicados à detecção de Spam. A escolha de modelos mais robustos, a adoção de técnicas de mitigação e a seleção apropriada da técnica de vetorização são aspectos fundamentais para garantir a segurança e confiabilidade desses sistemas. Pesquisas futuras devem explorar ataques mais sofisticados, estratégias de defesa e a análise do envenenamento em diferentes modelos e arquiteturas.

## 7. Agradecimentos

Este trabalho foi realizado com apoio do CNPq pelo Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação. ID: 9579, Edital PROPe 10/2023.

Durante a elaboração deste artigo, utilizou-se a ferramenta ChatGPT (modelo GPT-4o) para auxiliar no aprimoramento e na correção textual. A ferramenta também foi empregada para revisar a tradução do resumo para o *abstract* em língua inglesa.

## Referências

Bassiouni, M., Ali, M., and El-Dahshan, E. A. (2018). Ham and spam e-mails classification using machine learning techniques. *Journal of Applied Security Research*, 13:315–331.

Bhidya, M. (2019). Utkml's twitter spam detection competition. Disponível em:`https://kaggle.com/competitions/twitter-spam`. Acesso em: 25 ago. 2023.

Biggio, B., Nelson, B., and Laskov, P. (2013). Poisoning attacks against support vector machines.

Bindu, P. V., Mishra, R., and Thilagam, P. S. (2018). Discovering spammer communities in twitter. *Journal of Intelligent Information Systems*, 51:503–527.

Chen, Q., Zhang, R., Zheng, Y., and Mao, Y. (2022a). Dual contrastive learning: Text classification via label-aware data augmentation.

Chen, X., Dong, Y., Sun, Z., Zhai, S., Shen, Q., Wu, Zhonghai, e.-V., Di Pietro, R., Jensen, C. D., and Meng, W. (2022b). Kallima: A clean-label framework for textual backdoor attacks. In *Computer Security – ESORICS 2022*, pages 447–466. Springer International Publishing.

Dada, E. G., Bassi, J. S., Chiroma, H., Abdulhamid, S. M., Adetunmbi, A. O., and Ajibuwa, O. E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6):e01802.

Derner, E. and Batistič, K. (2023). Beyond the safeguards: Exploring the security risks of chatgpt.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Hu, C. and Hu, Y.-H. F. (2020). Data poisoning on deep learning models. In *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 628–632.

Hui, L. and Belkin, M. (2021). Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks.

Islam, R. and Moushi, O. M. (2024). Gpt-4o: The cutting-edge advancement in multimodal llm.

Janiesch, C., Zschech, P., and Heinrich, K. (2021). Machine learning and deep learning. *Eletronic Markets*, 31:685–695.

Li, J., Yang, Y., Wu, Z., Vydiswaran, V. G. V., and Xiao, C. (2023). Chatgpt as an attack tool: Stealthy textual backdoor attack via blackbox generative model trigger.

Lichman, M. (2017). Youtube spam collection data set. Disponível em: `https://archive.ics.uci.edu/ml/datasets/YouTube+Spam+Collection`. Acesso em: 23 ago. 2023.

NaliniPriya, G. and Asswini, M. (2015). A survey on vulnerable attacks in online social networks. *International Confernce on Innovation Information in Computing Technologies*, pages 1–6.

Rao, S., Verma, A. K., and Bhatia, T. (2021). A review on social spam detection: Challenges, open issues, and future directions. *Expert Systems with Applications*, 186.

Team, G. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv e-prints*, page arXiv:2403.05530.

Utaliyeva, A., Pratiwi, M., Park, H., and Choi, Y.-H. (2023). Chatgpt: A threat to spam filtering systems. pages 1043–1050.

Wang, Q., Ma, Y., Zhao, K., and Tian, Y. (2020). A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, pages 1–26.

Yerlikaya, F. A. and Şerif Bahtiyar (2022). Data poisoning attacks against machine learning algorithms. *Expert Systems with Applications*, 208:118101.

Zhang, X. and Ghorbani, A. A. (2020). An overview of online fake news: Characterization, detection, and discussion. *Information Processing  Management*, 57.