Uma Nova Abordagem para Detecção de Cabeçalhos SMTP Falsos usando Aprendizado Profundo e Geração de Dados Sintéticos

Patrick M. Tavares[1], **Dalbert M. Mascarenhas**[1]

[1]Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ
Petrópolis - RJ - Brasil

ptkmoralestavares@gmail.com, dalbertmm@yahoo.com.br

Abstract. *This work proposes an new approach for detecting anomalous e-mail headers, focusing on phishing, spam, and legitimate messages. A Multilayer Perceptron (MLP) is used for classification, and a Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) is applied to generate synthetic data. The Gumbel Softmax function simulates features from imbalanced datasets, and statistical tests evaluate the quality of the generated data. Ray Tune is used to optimize model hyperparameters. Results show that the proposed approach improves accuracy and generalization in e-mail header threat detection.*

Resumo. *Este trabalho propõe uma nova abordagem para a detecção de cabeçalhos de e-mail anômalos, com foco em mensagens de* phishing, spam *e legítimas. Utilizamos um Perceptron Multicamadas (MLP) para classificação e uma Rede Adversária Generativa com Gradiente Penalizado (WGAN-GP) para geração de dados sintéticos. A técnica Gumbel Softmax é empregada para simular características de conjuntos de dados desbalanceados, e os dados gerados são avaliados por testes estatísticos. O Ray Tune é utilizado para otimização dos hiperparâmetros do modelo. Os resultados demonstram que a abordagem proposta melhora a acurácia e a capacidade de generalização na detecção de ameaças em cabeçalhos de e-mail.*

1. Introdução

A falsificação de cabeçalhos no Protocolo Simples de Transferência de Correio (SMTP) é uma técnica recorrente em campanhas de *phishing* e disseminação de *spam*, explorando a ausência de mecanismos nativos de autenticação e integridade do protocolo [Kulkarni et al. 2024, Greco et al. 2024]. Cabeçalhos SMTP falsos podem enganar mecanismos tradicionais de detecção e comprometer a segurança de sistemas de e-mail [Kaushik et al. 2024, Gupta et al. 2024, Luo et al. 2025]. Essa vulnerabilidade, somada ao crescimento das ameaças cibernéticas baseadas em engenharia social, torna a detecção automatizada de anomalias em cabeçalhos uma tarefa crítica para a cibersegurança [Wosah et al. 2024, Shahila et al. 2024, Nazario 2006].

Modelos de aprendizado de máquina, como os Perceptrons de Multicamadas (MLP), têm se mostrado eficazes na detecção de padrões complexos em dados estruturados, como cabeçalhos de e-mail [Dhanalakshmi et al. 2024, Yilmaz et al. 2020]. No

entanto, a presença de conjuntos de dados desbalanceados, nos quais exemplos maliciosos são minoria, compromete a eficácia de modelos supervisionados. Nesse contexto, Redes Generativas Adversariais (GANs) surgem como aliadas promissoras, ao permitirem a geração de dados sintéticos que enriquecem classes minoritárias e aumentam a robustez dos classificadores [Guan 2023].

Este trabalho propõe uma abordagem híbrida que combina MLPs e GANs para a detecção de cabeçalhos SMTP falsos. O modelo MLP atua como classificador principal, enquanto a GAN é utilizada para gerar exemplos sintéticos de cabeçalhos maliciosos, simulando ataques sofisticados e aprimorando a capacidade de generalização do sistema. A técnica Gumbel-Softmax é empregada para permitir a geração diferenciável de dados discretos — característica fundamental no domínio de cabeçalhos categóricos [Maddison et al. 2016, Zhou et al. 2022].

A metodologia proposta conta com uma etapa de pré-processamento dedicada à extração e normalização de campos de cabeçalho, seguida pela otimização de hiperparâmetros utilizando a ferramenta Ray Tune [Liaw et al. 2018]. Os experimentos foram realizados com dois conjuntos de dados públicos, compostos por e-mails legítimos, *spam* e *phishing* [Cormack and Lynam 2005, Nazario 2006], representados por atributos ternários extraídos dos cabeçalhos. A avaliação do desempenho foi conduzida por meio de métricas padrão, como acurácia, precisão, *recall* e *F1-score*, além de testes estatísticos como o *Classifier Two Sample Test* e o *Maximum Mean Discrepancy* para validação dos dados sintéticos [Lopez-Paz and Oquab 2018, Gretton et al. 2012].

Os resultados apontam que a integração entre MLP e GAN não apenas melhora a acurácia na detecção de cabeçalhos SMTP falsos, mas também aumenta a robustez do sistema contra variações e manipulações adversariais. Essa abordagem representa uma evolução significativa em relação às técnicas tradicionais de validação de cabeçalhos, oferecendo uma solução escalável e adaptável a ambientes diversos.

O restante deste artigo está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A Seção 3 descreve a metodologia proposta. A Seção 4 apresenta os resultados experimentais e sua análise. Por fim, a Seção 5 traz as conclusões e direções futuras.

2. Trabalhos Relacionados

A detecção de e-mails maliciosos tem sido amplamente explorada na literatura, com foco em técnicas de aprendizado de máquina, análise de texto e, mais recentemente, geração de dados sintéticos. Este trabalho se diferencia por abordar exclusivamente a análise de cabeçalhos SMTP com um modelo supervisionado baseado em *Multilayer Perceptron* (MLP), aliado à geração de amostras sintéticas via WGAN-GP com Gumbel-Softmax.

Bountakas et al. [Bountakas et al. 2021] investigam a eficácia de técnicas de *Processamento de Linguagem Natural* (NLP) para detecção de *phishing*, comparando TF-IDF, Word2Vec e BERT em conjunto com modelos como RF, DT, NB e LR. Embora tratem do problema do desbalanceamento de dados, não aplicam geração de amostras sintéticas e ignoram completamente os metadados dos cabeçalhos SMTP.

Karim et al. [Karim et al. 2020] propõem uma abordagem não supervisionada baseada em *clustering*, também utilizando cabeçalhos de e-mails. Embora compartilhem o

foco nos metadados, o trabalho não emprega aprendizado supervisionado nem geração de dados. Além disso, a redução de dimensionalidade é realizada de forma automática, enquanto este artigo utiliza 61 *features* discretas manualmente selecionadas e interpretáveis.

AbdulNabi e Yaseen [AbdulNabi and Yaseen 2021] aplicam BERT e BiLSTM para classificação de e-mails *spam*/*ham*, alcançando altas taxas de acurácia. A principal diferença está no foco no corpo textual e na ausência de mecanismos de balanceamento. O presente trabalho, por outro lado, atua sobre os cabeçalhos e implementa uma GAN para geração de dados sintéticos representativos.

Franchina et al. [Franchina et al. 2021] usam heurísticas e mineração de texto para detectar *phishing*, baseando-se na análise de conteúdo e remetentes suspeitos. Embora compartilhem o objetivo de detectar fraudes por e-mail, sua abordagem é textual e baseada em regras, sem o uso de aprendizado profundo ou geração de dados.

Beaman e Isah [Beaman and Isah 2022] propõem um modelo para detecção de anomalias baseado em 94 *features* de cabeçalhos, aplicando tanto classificadores binários como *One-Class SVM*. Embora próximos em escopo, não utilizam técnicas generativas para balanceamento de classes, diferentemente da abordagem híbrida proposta aqui.

Em síntese, os trabalhos revisados apresentam avanços importantes na classificação de e-mails maliciosos. Porém, diferentemente do presente trabalho, não combinam a análise de metadados de cabeçalhos SMTP com técnicas de aprendizado profundo e geração de dados sintéticos para tratar do desbalanceamento e aumentar a robustez da detecção.

3. Metodologia

A abordagem baseia-se na extração de características ternárias a partir dos cabeçalhos de e-mails e, em seguida, aplica-se um modelo de aprendizado supervisionado para classificar os e-mails como legítimos ou fraudulentos. A solução é composta por diversas etapas que englobam desde o pré-processamento dos dados até a classificação final, conforme é detalhado na Figura 1.

3.1. Pré-processamento de Dados

O pipeline de pré-processamento de dados consiste em múltiplas etapas que transformam dados brutos de e-mails em vetores de características adequados para modelos de aprendizado de máquina. O processo envolve a organização do conjunto de dados, a extração de características relevantes e a preparação dos dados para treinamento. A Figura 2 ilustra o pipeline completo de pré-processamento.

3.1.1. Organização de Dados

Para realizar a detecção de cabeçalhos SMTP falsos, foi necessário construir um conjunto de dados abrangente contendo e-mails legítimos (*ham*), fraudulentos (*phishing*) e de *spam*. Essa construção exige uma organização dos dados para garantir que os diferentes tipos de e-mails estejam devidamente categorizados e acessíveis para análise.

Nesse contexto, a etapa inicial do pré-processamento, implementada no arquivo `dataOrganizer.py`, desempenha um papel fundamental ao estruturar os arquivos de

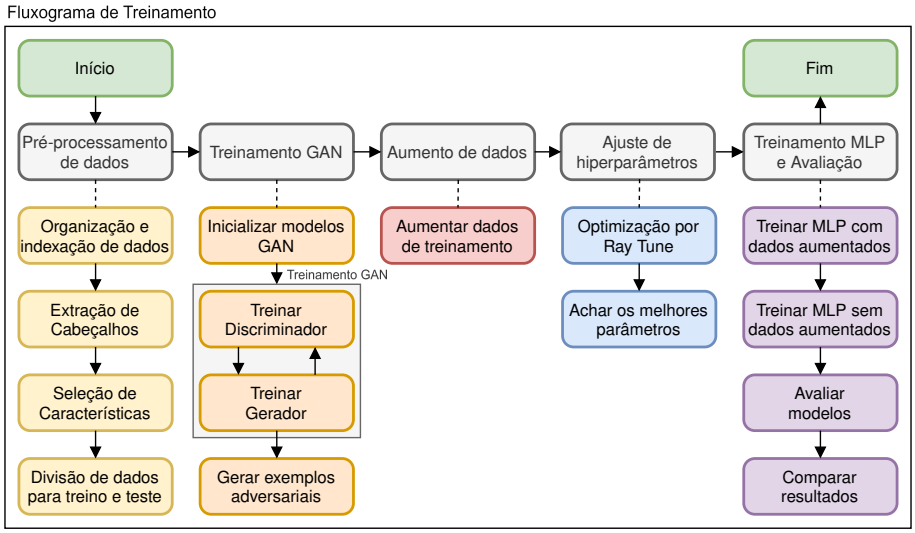

Figura 1. Fluxograma do treinamento. As etapas estão destacadas por cores: azul para o processamento de dados, roxo para a utilização de modelos e laranja para a avaliação.

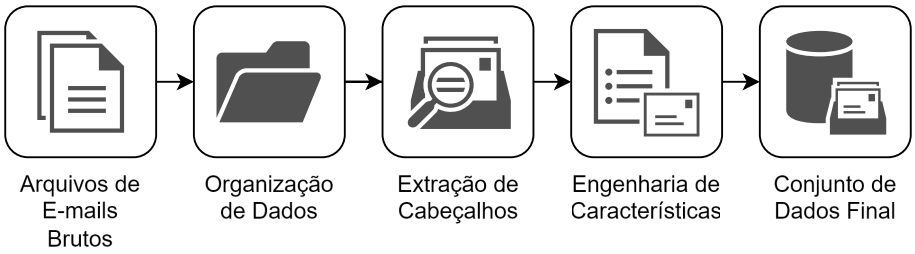

Figura 2. Pipeline de preprocessamento de dados.

e-mails brutos. O processo inclui a criação de uma estrutura de diretórios unificada para acomodar todos os tipos de e-mails, como *ham*, *spam* e *phishing*. Os e-mails *spam* e *ham* do conjunto de dados TREC07P [Cormack and Lynam 2005] são copiados para o diretório designado, enquanto os e-mails *phishing* [Nazario 2006] são processados a partir de arquivos TXT separados. Além disso, é gerado um arquivo de índice que mapeia os caminhos dos e-mails para seus respectivos rótulos, facilitando o uso do conjunto de dados em etapas posteriores da análise.

3.1.2. Extração de cabeçalhos e características

Os cabeçalhos SMTP dos e-mails coletados foram processados para extrair características relevantes para modelos de aprendizado de máquina. A Figura 3 ilustra este processo.

Utilizou-se a biblioteca HeaderParser [II 2023] para processar 61 campos definidos em `HEADER_INFORMATION`, com atenção especial aos cabeçalhos 'received'. As características foram organizadas em categorias:

- Temporais: como `time_zone` e diferenças entre datas
- Dados ausentes: como `missing_mime-version` e `missing_dmarc`

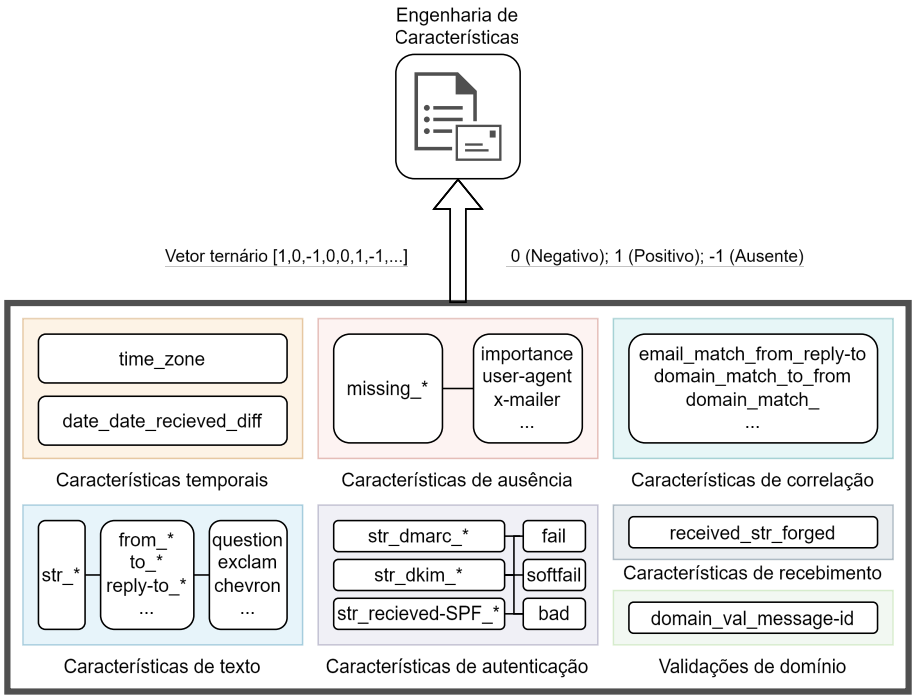

Figura 3. Diagrama de Engenharia de Características.

- Padrões de strings: detectando caracteres suspeitos em campos como 'From' e 'Message-ID'
- Autenticação: verificações SPF, DMARC e DKIM
- Validação de domínios: comparando campos como 'Message-ID' e 'From'

A função `get_email_info` processou os e-mails em 'latin_1', convertendo os dados em valores ternários (0/1/-1) que indicam a ausência, presença ou invalidade das informações. O pipeline final incluiu:

- Balanceamento de classes (*ham/phishing* ou *ham/spam*)
- Divisão treino/teste
- Armazenamento em cache
- Processamento paralelo com `ThreadPoolExecutor`

O resultado foram vetores normalizados contendo as 61 características por e-mail, prontos para a modelagem.

3.1.3. Treinamento e geração de exemplos com WGAN-GP

A arquitetura WGAN-GP proposta, ilustrada na Figura 4, consiste em duas redes principais: o Gerador e o Discriminador (Crítico). Ambas utilizam blocos residuais para facilitar o treinamento de redes profundas, permitindo um melhor fluxo do gradiente durante a retropropagação.

O Gerador processa um vetor latente de dimensão 512 através de uma camada linear inicial e dois blocos residuais consecutivos. Cada bloco residual inclui normalização de camada (*Layer Norm*), ativação ReLU, camadas lineares e uma conexão direta (*skip*

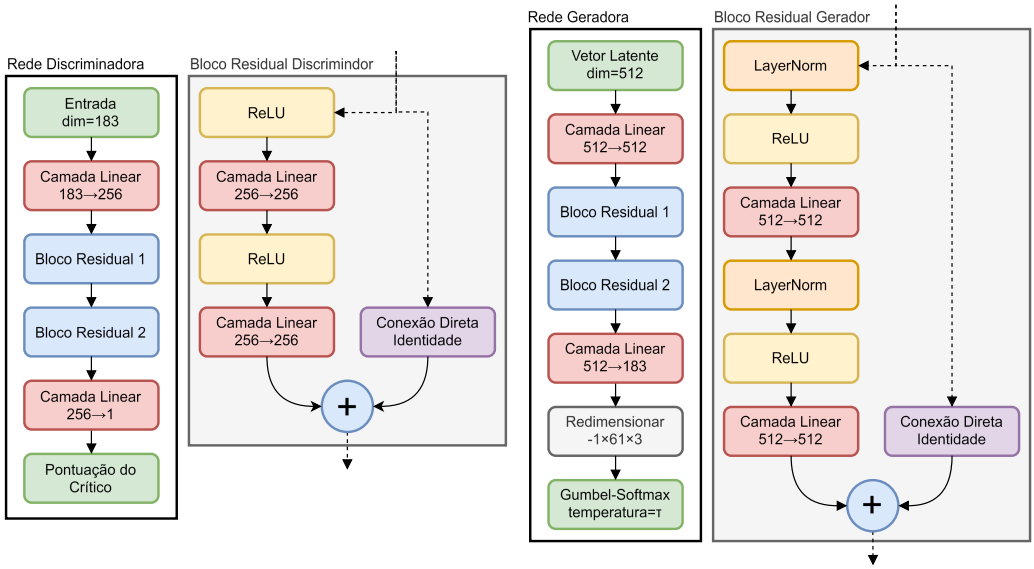

Figura 4. Arquitetura do Gerador e Discriminador deste trabalho.

connection) para evitar a degradação do gradiente [Gulrajani et al. 2017]. A saída passa por uma camada linear final, reduzindo a dimensão para 183, seguida por redimensionamento para o formato -1×61×3 e aplicação da função Gumbel-Softmax, que permite a geração de dados discretos (0, 1 e -1) correspondentes às *features* de ausência, presença ou invalidade das informações extraídas.

O Discriminador recebe dados (gerados ou reais) de dimensão 183, transformando-os para dimensão 256 através de uma camada linear. Em seguida, processa o sinal por dois blocos residuais similares aos do Gerador, mas sem normalização de camada, compostos por ativações ReLU, camadas lineares e conexões diretas. Uma camada linear final reduz a dimensão para 1, produzindo uma pontuação que avalia a autenticidade dos dados.

A interação entre Gerador e Discriminador permite aprimoramento mútuo, com o Gerador produzindo dados cada vez mais realistas e o Discriminador tornando-se mais eficaz na distinção entre dados reais e gerados. O Gradient Penalty na WGAN-GP garante estabilidade no treinamento e o cumprimento da condição de Lipschitz pelo Discriminador.

O treinamento, ilustrado na Figura 5, inicia-se com a movimentação das redes para o dispositivo adequado (GPU) e configuração dos otimizadores Adam e *schedulers* para ajuste dinâmico das taxas de aprendizado. O processo contempla carregar *checkpoints* prévios, quando disponíveis.

Durante cada época, a temperatura para a função Gumbel-Softmax é calculada, influenciando a geração de dados discretos. O treinamento divide-se em duas fases principais:

1. Treinamento do Crítico: repetido `n_critic` vezes por *batch*, inicia com a geração de amostras falsas a partir de ruído latente. O Discriminador avalia amostras falsas e reais, calcula-se o *gradient penalty* para garantir a condição de Lipschitz [Arjovsky et al. 2017], e atualiza-se os parâmetros do Discriminador a partir da

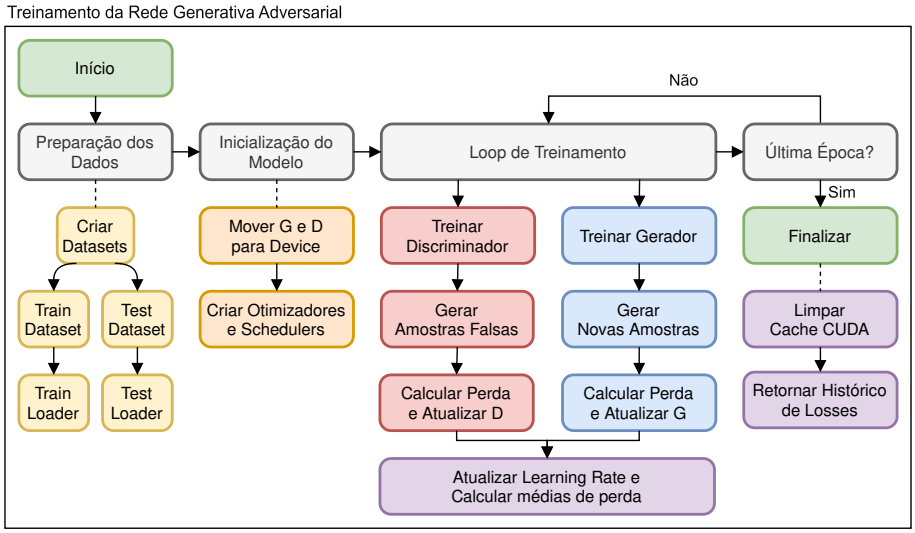

Figura 5. Fluxo de treinamento GAN deste trabalho.

perda Wasserstein.

2. Treinamento do Gerador: cria novas amostras falsas que são avaliadas pelo Discriminador, calcula a perda com base na capacidade de enganar o Discriminador e atualiza os parâmetros do Gerador. As taxas de aprendizado são ajustadas dinamicamente.

Ao final de cada época, calculam-se as médias das perdas para monitoramento e salvam-se *checkpoints* em intervalos predefinidos. Após o treinamento, o cache CUDA é liberado e o histórico de perdas é retornado para análise.

A WGAN-GP também permite gerar exemplos adversariais de cabeçalhos de e-mail, produzindo amostras sintéticas a partir de um vetor de ruído e temperatura controlada. Esta abordagem introduz maior diversidade ao conjunto de dados, aumentando a capacidade do modelo em identificar padrões variados e desconhecidos, contribuindo para uma classificação mais robusta de e-mails com cabeçalhos SMTP falsificados.

3.1.4. Treinamento e definição de hiperparâmetros da MLP

A Rede Neural de Multicamadas (MLP) proposta para classificação de cabeçalhos anômalos utiliza uma estrutura profunda com blocos residuais e técnicas de regularização para garantir uma generalização eficiente. O modelo recebe como entrada as 61 *features* extraídas dos cabeçalhos de e-mails e produz uma saída binária que classifica os dados como *phishing* vs *ham* ou *spam* vs *ham*.

A arquitetura, ilustrada na Figura 6, inicia com uma camada linear que transforma a dimensão de entrada para 512, seguida por dois blocos residuais. Cada bloco residual contém ativações ReLU, camadas lineares, *dropout* para prevenção de *overfitting*, e conexões diretas (*skip connections*) que preservam a informação inicial e facilitam o fluxo do gradiente. Após os blocos residuais, uma camada linear final reduz a dimensão para 2, correspondendo às classes de saída.

O modelo incorpora regularização L1, que promove esparsidade nos pesos, e

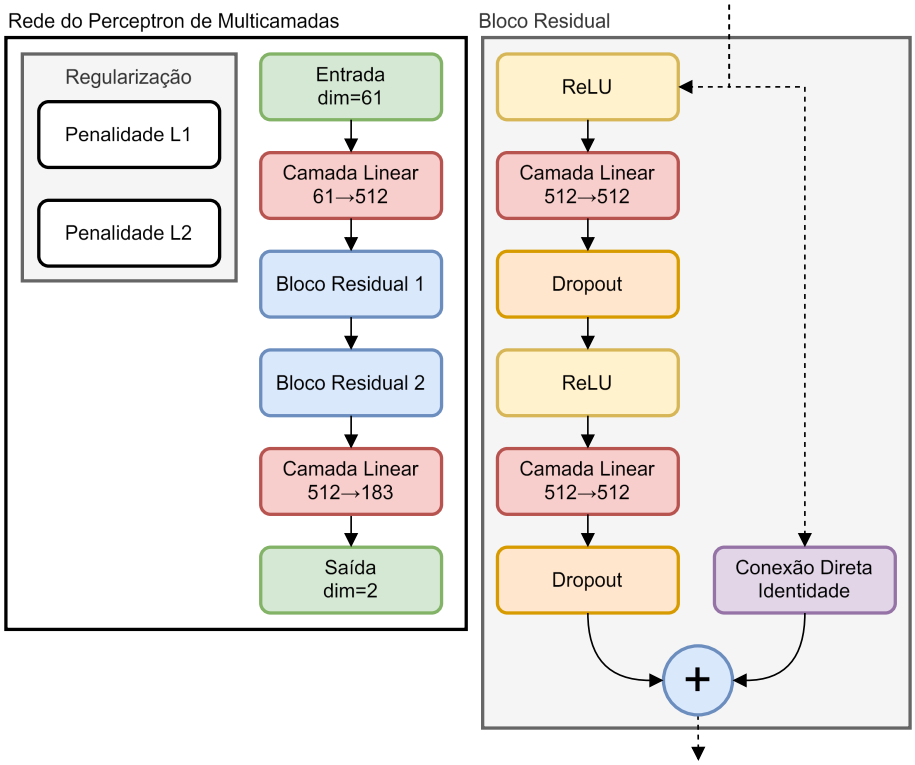

Figura 6. Arquitetura do perceptron de multicamadas deste trabalho.

regularização L2, que controla a magnitude dos pesos, prevenindo o *overfitting* e melhorando a generalização para dados não vistos.

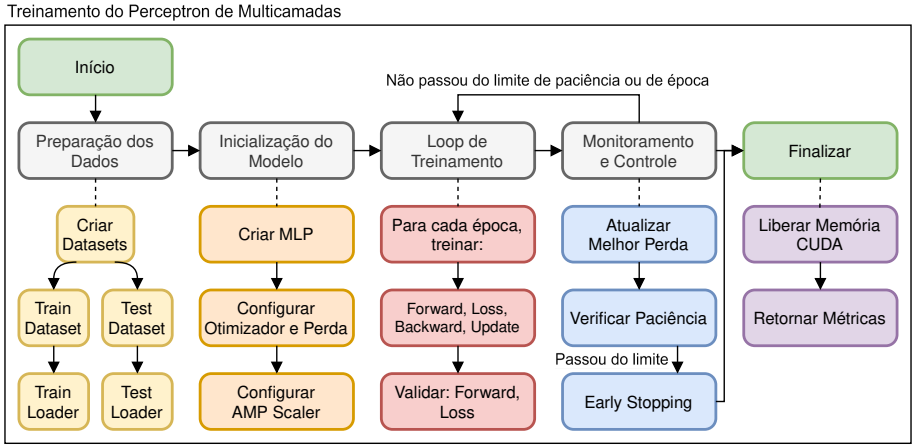

Figura 7. Fluxo de treinamento do perceptron de multicamadas deste trabalho.

O fluxo de treinamento, demonstrado na Figura 7, inicia com a preparação dos dados em *DataLoaders* e a inicialização do modelo, da função de perda e do otimizador. Utiliza-se AMP (*Automatic Mixed Precision Scaler*) para otimizar o uso de memória e acelerar o treinamento sem comprometer a precisão.

O treinamento ocorre em um loop de épocas com fases de treino e validação. Na fase de treino, com o modelo em modo de treinamento, os gradientes são zerados, o *AutoCast* CUDA é ativado para precisão mista, e executa-se o *forward pass*, cálculo de

loss (incluindo penalidades L1 e L2), *backward pass* e atualização de parâmetros. Na fase de validação, o modelo é colocado em modo de avaliação para processar dados de validação sem atualizar parâmetros.

Um sistema de monitoramento verifica se a *loss* de validação atual é a melhor já obtida. Se não houver melhoria após um número predefinido de épocas, o treinamento é interrompido (*Early Stop*), economizando recursos computacionais. No *Ray Tune*, este mecanismo permite explorar mais combinações de hiperparâmetros em menos tempo ao interromper treinos ineficazes. Durante todo o processo, monitoram-se as *losses* de treino e validação.

A busca pelos melhores hiperparâmetros utiliza o Ray Tune, que explora sistematicamente diferentes valores para penalizações L1 e L2, dimensões dos neurônios nas camadas ocultas, taxa de aprendizado, número de épocas e nível de *dropout*. O algoritmo emprega amostragens randômicas, critérios de redução de espaço e estratégias de paralelização para avaliar múltiplas combinações simultaneamente [Liaw et al. 2018]. O agendador ASHAScheduler prioriza dinamicamente experimentos promissores e encerra antecipadamente casos subótimos.

Nas etapas finais, são treinados dois modelos distintos: um com o conjunto original de dados e outro com dados aumentados gerados pela WGAN-GP. Ambos passam por fases idênticas de ajuste de hiperparâmetros e treinamento, permitindo comparar diretamente o impacto do aumento de dados na detecção de cabeçalhos SMTP falsos. Esta análise comparativa é apresentada na Seção 4, destacando as contribuições de cada abordagem no aprimoramento do desempenho do programa.

4. Resultados

Os resultados foram obtidos em um ambiente de testes configurado com processador Ryzen 5 5600x, 32 GB de memória RAM DDR4, e uma placa de vídeo NVIDIA RTX 4070 com 12 GB de VRAM. O software utilizado incluiu Windows 11 24H2, Python 3.11, PyTorch 2.5.1, RayTune 2.41.0 e Numpy 2.2.2.

4.1. Conjuntos de Dados

Foram analisadas 78.172 amostras, distribuídas em 25.220 e-mails classificados como *ham*, 50.199 como *spam* e 2.753 como *phishing*. Os dados foram divididos em 75% para treinamento e 25% para teste.

4.2. Métricas de Avaliação

Para a GAN, utilizou-se o *Classifier Two Sample Test* (C2ST), o *Maximum Mean Discrepancy* (MMD) e o valor-P. Para o MLP, foram empregadas métricas como precisão, *recall*, *F1-score* e acurácia, além de curvas de *loss* e matriz de confusão.

4.3. Resultados da Geração de Cabeçalhos Falsos com GAN

A geração de dados sintéticos foi realizada utilizando uma arquitetura *Wasserstein GAN with Gradient Penalty* (WGAN-GP) combinada com *Gumbel Softmax*. Os resultados do treinamento podem ser observados nas Figuras 9 e 8.

Para os dados de *phishing*, obteve-se uma acurácia C2ST de 48,33%, MMD de 0,0008 e valor-P de 0,1740. Para os dados *ham*, os resultados foram similares: acurácia

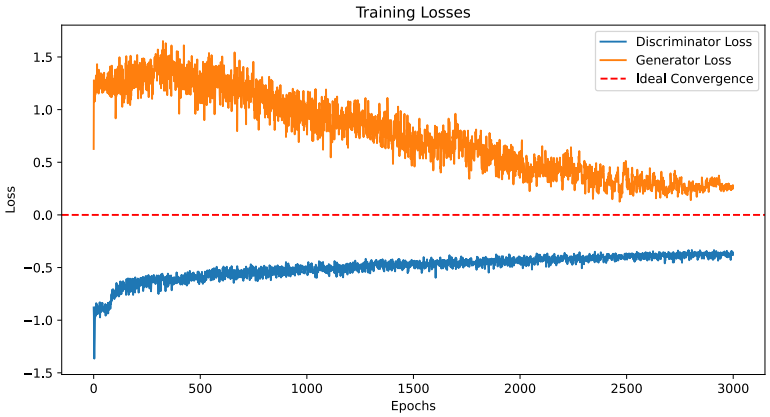

Figura 8. Valor de perda do Gerador e Discriminador na convergência para dados *ham* e *phishing*.

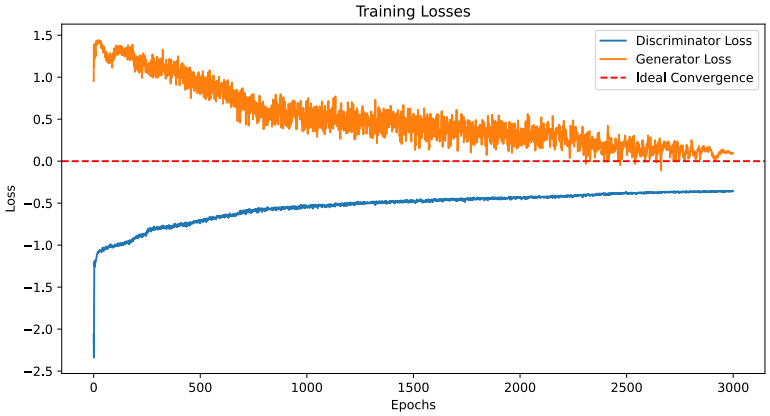

Figura 9. Valor de perda do Gerador e Discriminador na convergência para dados *ham* e *spam*.

C2ST de 49,33%, MMD de 0,0010 e valor-P de 0,07906. As distribuições de características dos dados originais e aumentados podem ser observadas na tabela 1, com padrões equivalentes para *ham/spam*.

	Treino e Aumentado Ham			Treino|Aumentado Phishing		
Feature	-1	0	1	-1	0	1
time_zone	0	5459	13396	51|384	121|963	1952|17508
date_comp_date_received	17	2515	16323	191|1404	306|2877	1627|14574
missing_*	—	—	—	—	—	—
*mimeversion	0	16370	2485	0|3	2109|18749	15|103
*xmailer	0	4400	14455	0|18	156|1249	1968|17588
*listunsubscribe	0	14978	3877	0|0	84|676	2040|18179
*xmailmanversion	0	9994	8861	0|0	0|0	2124|18855
*references	0	8491	10364	0|0	67|567	2057|18288

Continua na próxima página

Tabela 1 – *Continuação da página anterior*

	Treino e Aumentado			Treino|Aumentado		
	Ham			Phishing		
Feature	-1	0	1	-1	0	1
*useragent	0	4853	14002	0|0	18|123	2106|18732
*receivedspf	0	3364	15491	0|4	769|6983	1355|11868
*xoriginalto	0	4824	14031	0|0	0|14	2124|18841
*domainkeysignature	0	1498	17357	0|14	830|7266	1294|11575
*importance	0	876	17979	0|16	50|339	2074|18500
*dmarc	0	18855	0	0|0	2124|18855	0|0
str_*	—	—	—	—	—	—
*contentencoding_empty	0	0	18855	0|0	0|0	2124|18855
*from_question	0	18449	406	0|114	1950|17556	174|1185
*from_exclam	0	18852	3	0|0	2120|18848	4|7
*from_chevron	0	2541	16314	0|0	80|623	2044|18232
*to_chevron	0	10029	8826	0|1	1503|13618	621|5236
*to_undisclosed	0	18855	0	0|0	2117|18848	7|7
*to_empty	0	0	18855	0|0	0|0	2124|18855
*messageID_dollar	0	17077	1778	0|9	2088|18598	36|248
*returnpath_bounce	0	7500	11355	0|9	1927|17482	197|1364
*returnpath_empty	0	0	18855	0|0	0|0	2124|18855
*replyto_question	0	18854	1	0|13	2094|18557	30|285
*receivedSPF_bad	0	18855	0	0|0	2124|18855	0|0
*receivedSPF_softfail	0	18855	0	0|1	2029|18088	95|766
*receivedSPF_fail	0	18855	0	0|0	2078|18475	46|380
*contenttype_texthtml	0	17418	1437	0|0	1373|12315	751|6540
*precedence_list	0	7702	11153	0|1	2117|18821	7|33
*dmarc_bad	0	18855	0	0|0	2124|18855	0|0
*dmarc_softfail	0	18855	0	0|0	2124|18855	0|0
*dmarc_fail	0	18855	0	0|3	2107|18777	17|75
*dkim_bad	0	18855	0	0|0	2124|18855	0|0
*dkim_softfail	0	18855	0	0|0	2124|18855	0|0
*dkim_fail	0	18855	0	0|1	2116|18765	8|89
received_str_forged	0	18726	129	0|0	2115|18765	9|90
email_match_from_replyto	12436	4638	1781	1807|16092	197|1787	120|976
domain_val_messageid	1269	17126	460	404|3598	1720|15252	0|5
domain_match_*	—	—	—	—	—	—
*messageid_from	1270	5212	12373	429|3804	627|5515	1068|9536
*from_returnpath	1	12126	6728	37|272	334|2553	1753|16030
*messageid_returnpath	1269	12384	5202	423|3773	548|4765	1153|10317
*messageid_sender	8775	7309	2771	2094|18605	14|127	16|123
*messageid_replyto	13145	2190	3520	1897|16902	176|1516	51|437
*returnpath_replyto	12436	1395	5024	1806|16079	215|1916	103|860
*replyto_to	12532	2896	3427	1836|16360	262|2348	26|147
*to_inreplyto	11194	3953	3708	2090|18541	20|182	14|132
*errorsto_messageid	9057	7137	2661	2124|18855	0|0	0|0
*errorsto_from	8199	7572	3084	2124|18853	0|2	0|0

Continua na próxima página

Tabela 1 – *Continuação da página anterior*

	Treino e Aumentado Ham			Treino|Aumentado Phishing		
Feature	-1	0	1	-1	0	1
*errorsto_sender	8461	0	10394	2124|18855	0|0	0|0
*errorsto_replyto	14612	534	3709	2124|18854	0|1	0|0
*sender_from	7906	8041	2908	2084|18582	17|89	23|184
*references_replyto	17186	1122	547	2100|18638	24|203	0|14
*references_inreplyto	11688	85	7082	2083|18495	0|0	41|360
*references_to	10803	4230	3822	2067|18346	41|370	16|139
*from_replyto	12436	1888	4531	1807|16074	155|1446	162|1335
*to_from	116	15416	3323	154|1409	1564|14098	406|3348
*to_messageid	1382	14446	3027	514|4624	1496|13345	114|886
*to_received	17537	7	1311	1337|11823	2|29	785|7003
*replyto_received	18028	827	0	2011|17848	113|1006	0|1
*from_received	18855	0	0	2124|18855	0|0	0|0
*returnpath_received	18855	0	0	2124|18854	0|0	0|1

Tabela 1. Distribuição de Features - Valores Absolutos (Treino vs Aumentado)

A capacidade da GAN em manter a estrutura discreta dos dados (-1, 0, 1) foi fundamental para preservar a semântica das características dos cabeçalhos, demonstrando que a abordagem proposta foi capaz de gerar dados sintéticos de alta qualidade que mantêm as características essenciais dos cabeçalhos de e-mail originais.

4.4. Resultados da Classificação com MLP

Os resultados obtidos na classificação de cabeçalhos utilizando o modelo MLP com dados aumentados demonstraram excelente desempenho tanto na distinção entre *phishing* vs *ham* quanto *spam* vs *ham*. A Figura 11 apresenta o treinamento e matriz de confusão para a classificação de *phishing*, enquanto a Figura 10 demonstra aspectos semelhantes para *spam*.

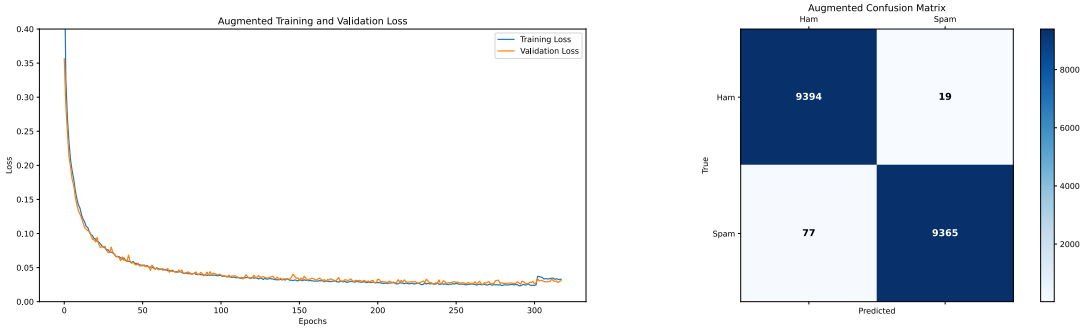

Figura 10. Curva de perdas e matriz de confusão do treinamento MLP aumentado entre ham e spam.

Na classificação entre *phishing* e *ham*, o modelo alcançou 98,81% de acurácia, superando ligeiramente os resultados de [Beaman and Isah 2022]. De 3.461 cabeçalhos legítimos, 3.417 foram classificados corretamente, com apenas 44 falsos positivos. Dos

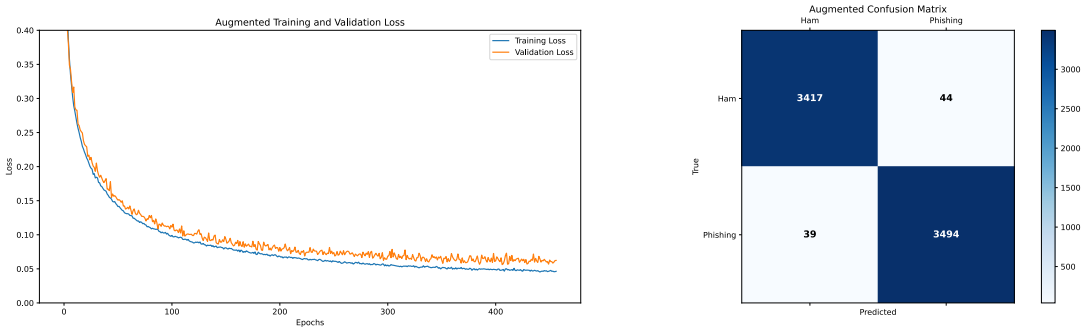

Figura 11. Curva de perdas e matriz de confusão do treinamento MLP aumentado entre ham e phishing.

3.533 cabeçalhos de *phishing*, 3.494 foram identificados corretamente. O modelo demonstrou equilíbrio entre precisão e *recall*, alcançando 0,9881 para ambos e para o *F1-score*.

Na classificação entre *spam* e *ham*, o modelo atingiu 99,49% de acurácia. Dos 9.413 cabeçalhos legítimos, 9.394 foram classificados corretamente, com apenas 19 falsos positivos. Para os 9.442 de *spam*, 9.365 foram identificados corretamente, com 77 falsos negativos.

As curvas de perda demonstram convergência estável, sem sinais significativos de *overfitting*. As limitações observadas concentram-se nos casos de falsos positivos e negativos que, embora poucos, podem representar riscos em aplicações reais. O processo de otimização via RayTune foi fundamental para este desempenho, permitindo equilíbrio entre generalização e precisão.

4.5. Impacto da Integração entre MLP e GAN

A análise comparativa entre o modelo MLP isolado e o integrado com WGAN-GP revela nuances interessantes. As Figuras 12 e 13 apresentam os resultados do MLP isolado.

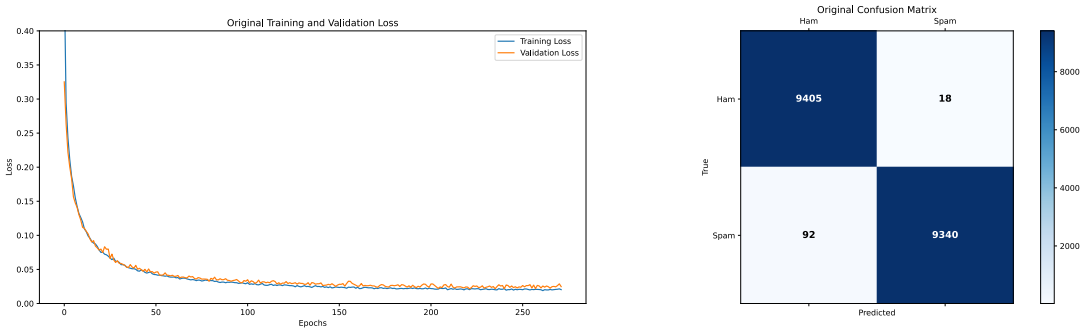

Figura 12. Curva de perdas e matriz de confusão do treinamento MLP isolado entre *ham* e *spam*.

Na classificação de *phishing*, o MLP isolado alcançou 98,77% de acurácia, enquanto Beaman e Isah [Beaman and Isah 2022] obtiveram aproximadamente 97,86%. A integração com WGAN-GP melhorou a acurácia para 98,81%, com redução de falsos negativos.

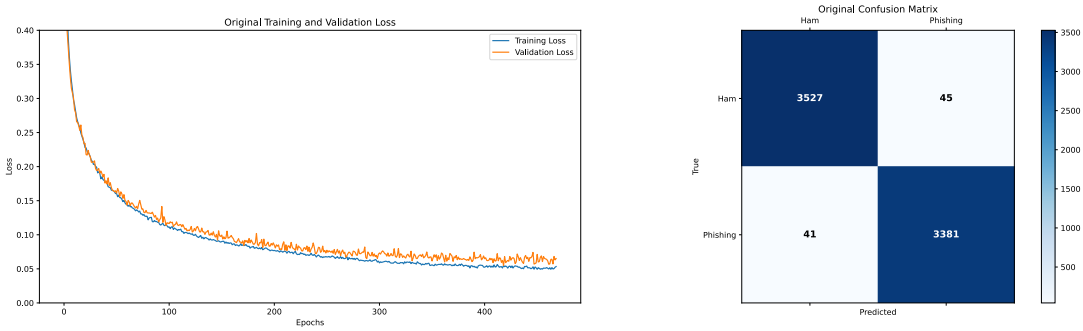

Figura 13. Curva de perdas e matriz de confusão do treinamento MLP isolado entre *ham* e *phishing*.

Métrica	*Phishing* vs *Ham*			*Spam* vs *Ham*		
	Original	Aumentado	Beaman	Original	Aumentado	Beaman
Acurácia	98,77%	98,81%	97,86%	99,42%	99,49%	99,56%
Precisão	98,77%	98,81%	96,32%	99,42%	99,49%	99,68%
Recall	98,77%	98,81%	99,15%	99,42%	99,49%	99,66%
F1-score	98,77%	98,81%	97,72%	99,42%	99,49%	99,56%
FP	45	44	-	18	19	-
FN	41	39	-	92	77	-

Tabela 2. Comparação de desempenho entre MLP isolado, MLP com dados aumentados e resultados da literatura.

Para classificação de *spam*, o impacto da integração foi mais notável. O MLP isolado atingiu 99,42% de acurácia, enquanto [Beaman and Isah 2022] alcançou aproximadamente 99,56%. Com a WGAN-GP, a acurácia aumentou para 99,49%, com redução significativa nos falsos negativos de 92 para 77 casos.

A integração com WGAN-GP proporcionou melhorias sutis nas métricas gerais, especialmente na redução de falsos negativos e na convergência mais consistente durante o treinamento, aspectos particularmente relevantes para aplicações práticas.

4.6. Discussão dos resultados

A combinação entre WGAN-GP com Gumbel Softmax e classificação via MLP resultou em um sistema eficaz para identificação de cabeçalhos anômalos. A avaliação da GAN revelou alta similaridade entre dados sintéticos e reais, com baixos valores de *Maximum Mean Discrepancy* e resultados positivos no *Classifier Two Sample Test*.

O MLP alcançou 98,81% de acurácia para *phishing* vs *ham* e 99,49% para *spam* vs *ham*. A integração com dados aumentados proporcionou redução relevante de falsos negativos, essencial para sistemas de detecção de ameaças.

Apesar dos resultados promissores, existem limitações, como a presença de falsos positivos e oscilações nas perdas da GAN durante o treinamento. O uso do RayTune

para a otimização de hiperparâmetros foi determinante para maximizar o desempenho, resultando em um treinamento consistente, sem sinais significativos de *overfitting*.

Conclui-se que a abordagem é eficaz para a detecção de cabeçalhos anômalos e apresenta um equilíbrio entre precisão e generalização. Melhorias futuras poderão incluir ajustes na arquitetura da GAN, além da incorporação de técnicas adicionais para o tratamento do desbalanceamento de classes.

4.7. Aplicabilidade e Viabilidade

Além do desempenho em métricas tradicionais, é importante discutir a aplicabilidade prática da abordagem proposta. A combinação entre WGAN-GP e MLP introduz um custo computacional considerável, especialmente durante a etapa de geração de dados sintéticos. O treinamento da WGAN-GP é notoriamente custoso, devido ao uso de penalidade de gradiente, blocos residuais profundos e à necessidade de múltimas iterações do discriminador por ciclo de treinamento. No ambiente experimental adotado (RTX 4070 com 12 GB de VRAM), o treinamento completo da GAN levou cerca de 4 horas por classe (*ham*/*spam* ou *ham*/*phishing*).

Em contrapartida, o processo de inferência do modelo MLP treinado é altamente eficiente. Após o treinamento, a classificação de novos cabeçalhos é realizada em tempo real, com latência média inferior a 5 milissegundos por amostra. Esse desempenho torna o sistema viável para integração em pipelines de detecção de ameaças em servidores de e-mail, sem comprometer a responsividade.

Outro ponto relevante é que a geração de dados sintéticos pode ser realizada de forma offline, antes do treinamento final do classificador. Isso permite que o *overhead* computacional associado à WGAN-GP seja confinado à fase de desenvolvimento ou revalidação periódica do modelo, sem impactar diretamente o uso em produção.

Portanto, embora a abordagem tenha um custo inicial elevado, sua estrutura modular permite separar as etapas intensivas em recursos da fase de inferência, tornando a solução viável para aplicações práticas em ambientes com restrições de tempo e desempenho.

5. Conclusão

Este trabalho propôs uma abordagem baseada na geração de dados sintéticos com WGAN-GP e Gumbel Softmax, integrada a um classificador MLP otimizado via RayTune, para detecção de cabeçalhos de e-mail anômalos. Os resultados obtidos demonstraram que a utilização de dados aumentados contribuiu para melhorias concretas nas métricas de desempenho, especialmente na redução de falsos negativos.

Na tarefa de detecção de *phishing*, a acurácia do modelo MLP aumentou de 98,77% para 98,81% com a inclusão de dados sintéticos, superando os 97,86% reportados por Beaman e Isah [Beaman and Isah 2022]. Embora o ganho percentual em acurácia tenha sido modesto, observou-se uma redução nos falsos negativos de 41 para 39 casos. Já na detecção de *spam*, o impacto foi mais expressivo: a acurácia subiu de 99,42% para 99,49%, com queda significativa nos falsos negativos, de 92 para 77 — uma redução de aproximadamente 16,3%.

Além das métricas, a integração com WGAN-GP resultou em uma convergência mais estável durante o treinamento, indicando maior robustez do modelo. Esses avanços, embora sutis em termos absolutos, são relevantes em aplicações reais, onde cada instância não detectada pode representar uma ameaça significativa.

Como perspectivas futuras, pretende-se explorar outras arquiteturas generativas, técnicas de balanceamento mais sofisticadas e cenários de validação em ambientes reais, com foco na adaptabilidade a novas variações de ataques e contextos de cibersegurança dinâmicos.

Referências

AbdulNabi, I. and Yaseen, Q. (2021). Spam email detection using deep learning techniques. *Procedia Computer Science*, 184:853–858. The 12th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 4th International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein gan.

Beaman, C. and Isah, H. (2022). Anomaly detection in emails using machine learning and header information.

Bountakas, P., Koutroumpouchos, K., and Xenakis, C. (2021). A comparison of natural language processing and machine learning methods for phishing email detection. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*, ARES '21, New York, NY, USA. Association for Computing Machinery.

Cormack, G. V. and Lynam, T. R. (2005). Trec 2007 public corpus. Permission is granted for research use only. Publishing the corpus or any part of it is prohibited.

Dhanalakshmi, R., Vijayaraghavan, N., Kumar, A., and Prathiba, B. S. B. (2024). Ai-based detection and analysis of phishing domains: Leveraging machine learning for enhanced cybersecurity. In *2024 International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–6. IEEE.

Franchina, L., Ferracci, S., and Palmaro, F. (2021). Detecting phishing e-mails using text mining and features analysis. In *Italian Conference on Cybersecurity*.

Greco, M., Chang, R., and Galdames, P. (2024). Educational phishing: An awareness campaign to learn how to detect phishing. In *2024 43rd International Conference of the Chilean Computer Science Society (SCCC)*, pages 1–5. IEEE.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773.

Guan, S. (2023). Performance analysis of convolutional neural networks and multilayer perceptron in generative adversarial networks. In *2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA)*, pages 817–821.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. (2017). Improved training of wasserstein gans.

Gupta, S., Pritwani, M., Shrivastava, A., Moharir, M., AR, A. K., et al. (2024). A comprehensive analysis of social engineering attacks: From phishing to prevention-tools,

techniques and strategies. In *2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI)*, pages 1–8. IEEE.

II, J. T. W. (2023). headerparser: argparse for mail-style headers. Biblioteca Python.

Karim, A., Azam, S., Shanmugam, B., and Kannoorpatti, K. (2020). Efficient clustering of emails into spam and ham: The foundational study of a comprehensive unsupervised framework. *IEEE Access*, 8:154759–154788.

Kaushik, N., Rathore, T. S., and Kumar, P. (2024). Email traceback: Securing systems from phishing and malicious link prevention. In *2024 1st International Conference on Advances in Computing, Communication and Networking (ICAC2N)*, pages 647–652. IEEE.

Kulkarni, M., Kumar, S., Panjwani, Y., Moharir, M., Kumar, A. A., Baskaran, E., et al. (2024). Mitigating email phishing: analytical framework, simulation models, and preventive measures. In *2024 10th international conference on communication and signal processing (ICCSP)*, pages 1459–1464. IEEE.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., and Stoica, I. (2018). Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*.

Lopez-Paz, D. and Oquab, M. (2018). Revisiting classifier two-sample tests.

Luo, E., Young, L., Ho, G., Afifi, M., Schweighauser, M., Katz-Bassett, E., and Cidon, A. (2025). Characterizing the networks sending enterprise phishing emails. In *International Conference on Passive and Active Network Measurement*, pages 437–466. Springer.

Maddison, C. J., Mnih, A., and Teh, Y. W. (2016). The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Nazario, J. (2006). Phishingcorpus homepage. Recuperado em Junho 2024.

Shahila, D. F. D., Rosi, A., Stephen, V., et al. (2024). Ai based phishing discrement for immense e-maildata. In *2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT)*, volume 1, pages 270–277. IEEE.

Wosah, P. N., Ali Mirza, Q., and Sayers, W. (2024). Analysing the email data using stylometric method and deep learning to mitigate phishing attack. *International Journal of Information Technology*, pages 1–12.

Yilmaz, I., Masum, R., and Siraj, A. (2020). Addressing imbalanced data problem with generative adversarial network for intrusion detection. In *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, pages 25–30.

Zhou, T., Wu, H.-T., Lu, H., Xu, P., and Cheung, Y.-M. (2022). Password guessing based on gan with gumbel-softmax. *Security and Communication Networks*, 2022(1):5670629.