UNIVERSIDADE DE SÃO PAULO
DEPARTAMENTO DE ENGENHARIA QUÍMICA

Uso de Inteligência Artificial no Desenvolvimento de Fármacos

Mateus Cioletti Senna

São Paulo
2024

MATEUS CIOLETTI SENNA

Uso de Inteligência Artificial no Desenvolvimento de Fármacos

Versão Original :

Texto do Trabalho de Conclusão de Curso
apresentado à Escola Politécnica da USP
Orientador: Prof. Dr. Ardson dos Santos Vianna Junior

São Paulo
2024

Resumo

Este estudo explora o uso de inteligência artificial no desenvolvimento e otimização de moléculas farmacêuticas em um contexto global. Primeiramente, o objetivo é explorar como conhecimento já é aplicado, trabalhos desenvolvidos e principais desafios encontrados na área. A partir disso, é feito um estudo de análise de viabilidade da aplicação dos conceitos em um caso real: desenvolvimento de drogas anti-histamínicas. O projeto envolve a criação de um modelo de Redes Neurais Gráficas para prever a probabilidade de atuação da molécula como antagonista dos receptores H1, principal mecanismo desse tipo de medicamento. O modelo construído mostrou boa capacidade de prever o mecanismo para o *dataset* de moléculas utilizado, prevendo 52% das moléculas ativas para o conjunto de teste.

Abstract

This study explores the use of artificial intelligence in the development and optimization of pharmaceutical molecules on a global scale. Initially, the aim is to explore how existing knowledge is applied, the work developed, and the main challenges encountered in the field. Subsequently, a feasibility study is conducted on the application of these concepts in a real case: the development of antihistamine drugs. The project involves creating a Graph Neural Network model to predict the likelihood of a molecule acting as an antagonist of H1 receptors, the primary mechanism of this type of medication. The model built demonstrated a good capacity to predict the mechanism for the dataset of molecules used, predicting 52% of active molecules in the testing set.

Sumário

1. Introdução

O desenvolvimento de fármacos é uma jornada complexa e multifacetada que se estende da descoberta de novas entidades químicas ou biológicas à aprovação regulatória e posterior comercialização. Este processo é crucial para introduzir novas terapias no mercado, proporcionando soluções para doenças desatendidas e melhorando os tratamentos existentes. No entanto, o desenvolvimento de medicamentos é frequentemente caracterizado por sua alta complexidade, enormes investimentos financeiros e uma taxa significativa de falhas (PAUFERRO, M. R. V. , 2020).

Atualmente, o setor enfrenta desafios substanciais. A complexidade dos mecanismos biológicos subjacentes às doenças muitas vezes torna a identificação de novos alvos terapêuticos uma tarefa hercúlea. Além disso, o aumento dos padrões regulatórios impõe um rigor adicional aos estudos clínicos, com a necessidade de demonstrar não apenas a eficácia, mas também a segurança e a qualidade dos novos fármacos.

Uma das maiores críticas ao processo de desenvolvimento de fármacos é a sua lentidão, por se tratar de um processo de alta complexidade e custo. Tradicionalmente, pode levar mais de uma década desde a concepção de uma nova molécula até ela se tornar um medicamento aprovado. Essa morosidade é exacerbada pela necessidade de extensos testes pré-clínicos e clínicos, exigidos para garantir a segurança e eficácia dos tratamentos. (PAUFERRO, M. R. V., 2020). O caminho é marcado por várias fases de avaliação, onde um candidato a medicamento deve demonstrar sucesso em testes in vitro, estudos em animais e múltiplas fases de ensaios clínicos em humanos. (PAUFERRO, M. R. V. , 2020)

Nesse contexto, a aplicação de técnicas de inteligência artificial mostra-se como uma ferramenta de alto potencial para otimização desse processo, podendo capturar padrões não facilmente identificáveis por análises humanas e processamento simultâneo de um elevado número de substâncias, o que poderia contribuir de forma significativa para a velocidade do desenvolvimento.

Neste trabalho é feito um estudo da aplicação de redes neurais gráficas para a classificação de moléculas que atuam como antagonistas do receptor h-1, mecanismo de funcionamento da maioria das drogas anti-histamínicas.

2. Objetivos

O objetivo do presente trabalho é fazer um estudo preliminar de como o uso de técnicas de inteligência artificial pode auxiliar o processo de desenvolvimento de novos fármacos. Para isso, primeiramente foi feito um estudo sobre recentes avanços no campo, diferentes abordagens que já são usadas na área em aplicações práticas, principais desafios encontrados e áreas a se desenvolver. Em seguida, o objetivo é realizar um estudo aplicado para construir um modelo capaz de auxiliar na descoberta de potenciais novas drogas anti-histamínicas, a partir da previsão da interação de moléculas como antagonistas do receptor H-1. Esse modelo poderia auxiliar o levantamento de potenciais moléculas no processo inicial do desenvolvimento de fármacos, levando a um maior direcionamento do processo.

3. Revisão Bibliográfica

3.1. Panorama sobre Inteligência Artificial no desenvolvimento de Fármacos

Embora ainda não possa ser considerada completamente consolidada e apresente diversas dificuldades na aplicação, a utilização de Inteligência Artificial na descoberta de fármacos está se tornando cada vez mais prevalente no mercado farmacêutico atual. A IA está rapidamente se tornando uma parte essencial do processo, à medida que as empresas farmacêuticas buscam inovações para acelerar o desenvolvimento de medicamentos e reduzir custos. A integração de IA está em um ponto de inflexão, avançando de experimentos e protótipos para aplicações práticas mais amplas (KIRKPATRICK, P, 2022).

Nesta seção, serão explorados trabalhos da literatura recente que foram influentes na abordagem do tema, aplicações que já são utilizadas no mercado e um estudo das principais dificuldades enfrentadas na área. Com isso, pode-se fazer inferências sobre como o trabalho pode trazer diferentes visões para o campo.

3.1.1. Estudos Relevantes

Seguindo a tendência de aplicações de conhecimentos de inteligência artificial como um todo, o uso de inteligência artificial no desenvolvimento é uma área que vem ganhando grande relevância no meio acadêmico, principalmente nos últimos anos. (KIRKPATRICK, P. , 2022) Há um enorme número de estudos recentes publicados na área, explorando diferentes aplicações e abordagens. Destes, vale mencionar alguns estudos de destaque em 3 diferentes áreas de aplicação:

Design de Moléculas

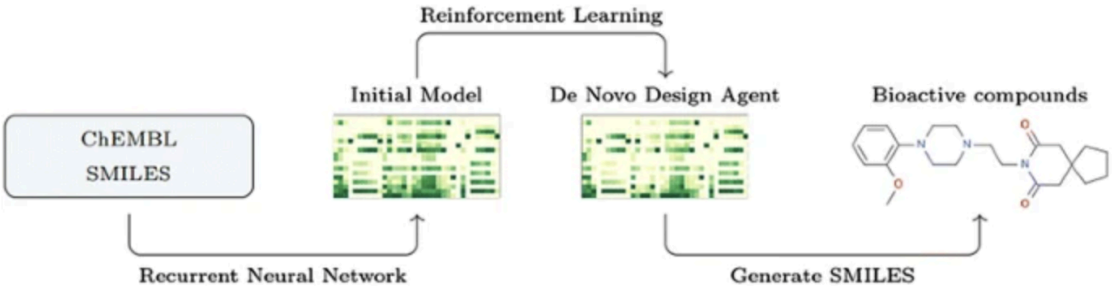
O design de moléculas . Um estudo importante no campo é "Molecular de-novo design through deep reinforcement learning" por Olivecrona et al. (2017). O estudo foca no desenvolvimento do REINVENT, um sistema que combina aprendizado profundo com aprendizado por reforço para gerar moléculas quimicamente viáveis com propriedades desejadas.

O REINVENT utiliza uma abordagem de aprendizado por reforço, onde um agente (o modelo de IA) aprende a realizar uma tarefa (gerar moléculas) de maneira que maximize uma recompensa (baseada nas propriedades desejadas das moléculas). O modelo emprega redes neurais recorrentes (RNNs), que são treinadas para gerar sequências de caracteres correspondentes a representações SMILES de moléculas.

A inovação central está na maneira como o modelo é treinado. Em vez de apenas aprender com um conjunto de dados pré-existente, o REINVENT é continuamente treinado e ajustado através de um ciclo de feedback, onde as moléculas geradas são avaliadas por sua adequação às propriedades desejadas e o modelo é ajustado para melhorar sua capacidade de gerar moléculas que atendam a esses critérios.

O REINVENT demonstrou ser capaz de gerar moléculas não apenas novas, mas também com propriedades específicas desejadas. Este trabalho representou um avanço significativo na descoberta de fármacos, pois oferece uma ferramenta poderosa para acelerar o processo de design de moléculas, permitindo que pesquisadores explorem espaços químicos com maior eficiência e criatividade.

Figura 1: Descrição da atuação do modelo REINVENT



Fonte: OLIVECRONA, M. et al. Molecular de-novo design through deep reinforcement learning. Journal of Cheminformatics, v. 9, n. 1, 4 set. 2017.

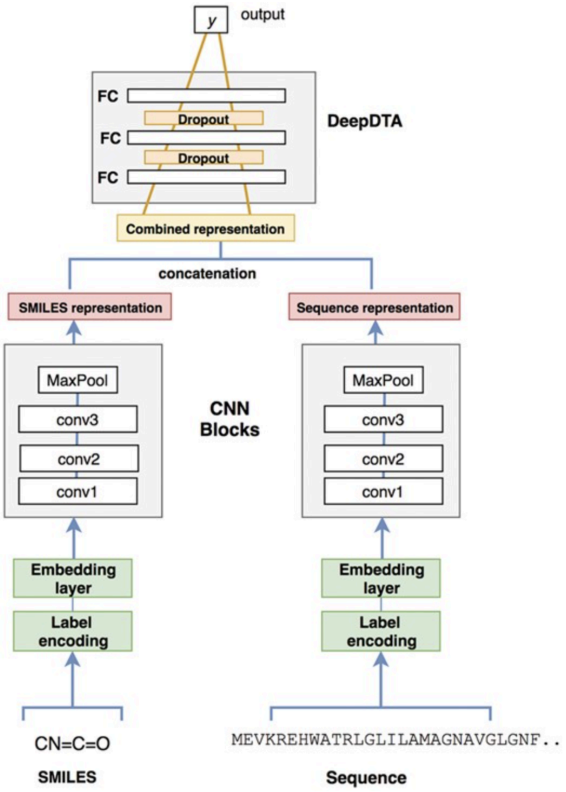
Identificação e Validação de Alvos

Ponto principal do caso de estudos do trabalho, a identificação eficaz de interações droga-alvo é um ponto chave para a descoberta de novos medicamentos. Tradicionalmente, este processo é intensivo em tempo e recursos, o que limita a velocidade da descoberta de novos fármacos.

A predição da afinidade de ligação entre drogas e alvos é complexa devido à natureza variável dos alvos biológicos e à diversidade estrutural dos compostos.

Nesse contexto, um estudo notável nessa aplicação é o estudo "DeepDTA: deep drug–target binding affinity prediction" de Öztürk, Özgür e Ozkirimli (2018), publicado na revista "Bioinformatics". O estudo introduziu o DeepDTA, um modelo baseado em redes neurais profundas para prever a afinidade de ligação droga-alvo. O DeepDTA foi projetado para processar sequências de aminoácidos de proteínas-alvo e estruturas químicas de compostos, superando desafios com dados não estruturados e heterogêneos.

Figura 2: Representação do funcionamento do modelo DeepDTA



Fonte: Öztürk, H., Özgür, A., & Ozkirimli, E. (2018). DeepDTA: deep drug–target binding affinity prediction. Bioinformatics, 34(17), i821-i829.

O modelo foi capaz de capturar características relevantes das sequências de proteínas e compostos, levando a previsões precisas de afinidade de ligação, superando significativamente os métodos tradicionais e alguns dos métodos computacionais contemporâneos em termos de precisão de predição.O modelo foi testado em diversos conjuntos de dados, demonstrando sua robustez e aplicabilidade em diferentes cenários.

Testes Pré-Clínicos e Toxicologia

A predição de toxicidade é outro tópico fundamental no desenvolvimento de fármacos. No contexto da área, o estudo "DeepTox: Toxicity Prediction using Deep Learning", Mayr et al. (2016), publicado na revista "Frontiers in Environmental Science", é um trabalho significativo na aplicação de técnicas de aprendizado profundo para a predição de toxicidade de compostos químicos. Este estudo representa um avanço importante na toxicologia computacional, uma área crítica no desenvolvimento de novos fármacos e na avaliação de segurança química.

A toxicidade é um fenômeno complexo, influenciado por múltiplos fatores e mecanismos biológicos. A capacidade de analisar grandes conjuntos de dados com variáveis complexas é fundamental para a predição eficaz da toxicidade. Nesse contexto, o estudo em questão utiliza redes neurais profundas (deep learning) para modelar e prever a toxicidade, modelo conhecido por sua habilidade em aprender representações complexas de dados, sendo particularmente adequado para lidar com a alta dimensionalidade e complexidade dos dados de toxicidade.

O estudo empregou o conjunto de dados Tox21, que inclui ensaios de toxicidade para mais de 10.000 compostos. O desafio Tox21 proporcionou uma plataforma para comparar diferentes métodos de predição de toxicidade, incluindo o modelo DeepTox. O DeepTox alcançou um desempenho superior na previsão de toxicidade em comparação com métodos tradicionais e outros modelos computacionais. Ele se destacou particularmente no desafio Tox21, demonstrando a viabilidade do deep learning na toxicologia computacional. (Mayr, A. , 2016)

3.1.2. Aplicações práticas no mercado

A aplicação de técnicas de inteligência artificial dentro da indústria farmacêutica vem se consolidando de forma significativa como ferramenta fundamental no desenvolvimento de fármacos. Além de grandes indústrias farmacêuticas comumente possuírem um departamento interno para tratar o tema, existem algumas empresas que se consolidaram formalmente nisso. Empresas como Atomwise, BenevolentAI e Recursion Pharmaceuticals estão na vanguarda

deste movimento, utilizando IA para modelar e prever a interação de moléculas com alvos biológicos.

Atomwise

Atomwise é uma empresa fundada em 2012 que se especializou em aplicar inteligência artificial para a descoberta de novos medicamentos. A Atomwise utiliza uma tecnologia de rede neural artificial conhecida como AtomNet para prever a eficácia de moléculas pequenas na modulação de proteínas específicas que podem estar associadas a doenças.

A rede é treinada em grandes *datasets* que incluem informações sobre a estrutura de proteínas e moléculas pequenas, assim como dados históricos sobre a eficácia de diferentes compostos. Isso permite que a AtomNet aprenda padrões complexos e identifique candidatos promissores para testes adicionais.

Um dos sucessos notáveis da Atomwise foi a identificação de dois medicamentos que poderiam ser potencialmente utilizados para tratar o vírus Ebola, uma realização que foi feita em colaboração com pesquisadores da Universidade de Toronto. A capacidade da empresa de identificar rapidamente compostos promissores para testes e desenvolvimento subsequente é um exemplo da aplicação prática de sua tecnologia (MARTIN, G. , 2016).

A empresa tem colaborado com instituições acadêmicas, empresas de biotecnologia e organizações farmacêuticas em todo o mundo. Essas parcerias buscam aplicar a tecnologia de IA da Atomwise para descobrir novos medicamentos para uma variedade de doenças, incluindo, mas não se limitando a, câncer, doenças infecciosas e distúrbios neurológicos (MARTIN, G. , 2016).

BenevolentAI

Assim como a Atomwise, a BenevolentAI é uma empresa de tecnologia que aplica inteligência artificial (AI) no desenvolvimento de medicamentos e no campo da pesquisa científica. Ela é uma das líderes no setor de biotecnologia, utilizando algoritmos avançados de aprendizado de máquina para acelerar a descoberta e o desenvolvimento de novos medicamentos.

BenevolentAI utiliza algoritmos de aprendizado de máquina e processamento de linguagem natural (PLN) para analisar grandes volumes de dados biomédicos e científicos. Isso inclui dados de ensaios clínicos, publicações científicas, bancos de dados de patentes e outros recursos relevantes (BUNTZ, B. , 2023).

A empresa muitas vezes se concentra em doenças complexas e de difícil tratamento, como doenças neurodegenerativas e câncer, onde a necessidade de novas terapias é mais urgente.

Enquanto a Atomwise concentra-se no uso de AI para previsão de estrutura molecular, especialmente no desenho de moléculas que possam se encaixar em alvos biológicos específicos (drug design), a BenevolentAI enfoca uma abordagem holística, combinando AI com biologia para descobrir novos medicamentos e alvos terapêuticos. Seu foco está em entender as doenças em um nível molecular e celular.

Exscientia

Assim como as duas empresas citadas anteriormente, a Excientia é uma empresa de enfoque em inteligência artificial (AI) na área da descoberta de medicamentos, conhecida por seu uso pioneiro de tecnologias automatizadas e aprendizado de máquina para revolucionar o processo de desenvolvimento de medicamentos (UK Research and Innovation, 2023).

Assim como a Atomwise, a empresa desenvolveu uma plataforma proprietária de AI. Esta combina técnicas de aprendizado de máquina, design de medicamentos e bioinformática para criar um processo de descoberta de medicamentos eficiente e eficaz. A plataforma da Exscientia permite a geração e otimização de moléculas candidatas a medicamentos, reduzindo o tempo e os custos associados ao desenvolvimento de medicamentos (UK Research and Innovation, 2023).

O enfoque principal da empresa é a aplicação de AI para desenvolver terapias mais personalizadas, visando melhorar a eficácia do tratamento e reduzir os efeitos colaterais.

3.2. Alergias e Drogas Anti-histamínicas

O foco do estudo de aplicação de caso do trabalho, uma alergia é uma resposta exagerada do sistema imunológico a substâncias que, na maioria das pessoas, são inofensivas. Essas

substâncias, conhecidas como alérgenos, podem incluir pólen, pelos de animais, certos alimentos, e até medicamentos. Quando uma pessoa alérgica entra em contato com um alérgeno, seu sistema imunológico reage de forma excessiva, desencadeando sintomas que podem variar de leves a graves (CLEVELAND CLINIC , 2020).

Inicialmente, o sistema imunológico identifica erroneamente um alérgeno inofensivo como uma ameaça e produz anticorpos específicos contra ele (Imunoglobulina E, ou IgE). Na reexposição ao mesmo alérgeno, os anticorpos IgE reconhecem e se ligam ao alérgeno. Essa ligação ocorre principalmente em células chamadas mastócitos, presentes em tecidos em todo o corpo. A ligação do alérgeno aos anticorpos IgE ativa os mastócitos, que liberam várias substâncias químicas, sendo a mais notável a histamina. A histamina liberada causa sintomas comuns da alergia, como inchaço, vermelhidão, coceira, espirros, e aumento da secreção mucosa, podendo causar sintomas mais graves como o bloqueio de respiração e até a morte (CRIADO, P. R , 2010) .

Nesse contexto, anti-histamínicos são drogas que atuam bloqueando os receptores de histamina, especialmente os receptores H1, encontrados nas células epiteliais e nos vasos sanguíneos. Ao bloquear esses receptores, os anti-histamínicos impedem que a histamina se ligue a eles, o que reduz os sintomas alérgicos (CRIADO, P. R , 2010) .

Os primeiros medicamentos dessa classe desenvolvidos, chamados de anti-histamínicos de primeira geração, desenvolvidos inicialmente na década de 1940, foram revolucionários em cumprir a função entretanto, apresentavam efeitos sedativos significativos devido à sua capacidade de atravessar a barreira hematoencefálica. Como uma resposta às limitações dos anti-histamínicos de primeira geração particularmente no que diz respeito aos seus efeitos colaterais sedativos, foram desenvolvidos os anti-histamínicos de segunda geração. O desenvolvimento destes medicamentos envolveu uma compreensão mais profunda da farmacologia dos receptores de histamina e a estrutura química das moléculas envolvidas. (CLEVELAND CLINIC, 2020) Exemplos de anti-histamínicos de segunda geração são a loratadina, epinastina, ebastina e cetirizina.

Apesar de terem resultados significativos com baixo índice de efeitos colaterais, os anti-histamínicos, tanto de primeira quanto de segunda geração, ainda apresentam limitações e precauções específicas. Por exemplo, se tratando de pacientes com glaucoma, especialmente o

glaucoma de ângulo fechado. Esta precaução decorre do mecanismo de ação desses medicamentos e de como eles podem afetar a fisiologia ocular. O uso desses medicamentos pode desencadear um ataque agudo de glaucoma de ângulo fechado, uma emergência oftalmológica que requer atenção médica imediata (WERNER, M , 2022).

3.3. Obtenção de Dados

Para o desenvolvimento do trabalho, necessita-se de uma grande quantidade de dados contendo diferentes atributos de moléculas. Para isso, se fará uso de repositórios online de dados químicos e de interação biológica. Existem alguns repositórios disponíveis, cada um com diferentes abordagens, fins e tipos de dados. Algumas das principais possibilidades são :

Pubchem

O PubChem é um repositório de química mantido pelo National Center for Biotechnology Information (NCBI), que faz parte dos Institutos Nacionais de Saúde dos EUA. Ele serve como uma fonte gratuita e acessível de informações sobre a estrutura química, propriedades, atividades biológicas e patentes (KIM, S. et al. , 2015). O banco de dados do PubChem é extenso e útil para pesquisadores, cientistas e profissionais em vários campos relacionados à química e à biologia.

Chembl

O ChEMBL é um banco de dados bioinformático mantido pelo European Bioinformatics Institute (EBI), parte do European Molecular Biology Laboratory (EMBL). Ele é focado em dados de medicamentos e é amplamente utilizado para pesquisar informações sobre a relação entre compostos bioativos e suas propriedades biológicas (GAULTON, A. et al. , 2016) .

Mais completo e robusto que o pubChem no que diz respeito a interações biológicas e aplicações medicamentosas, o ChemBl possui API aberta e gratuita, com mais de 20 milhões de atividades biológicas de moléculas reportadas de diferentes fontes de conhecimento.

DrugBank

A mais abrangente e aprofundada entre as citadas para o estudo de interações biológicas,

o DrugBank é um recurso bioinformático único usado amplamente em pesquisa farmacêutica e medicina. Assim como o Chembl e o Pubchem, reúne informações detalhadas sobre drogas e medicamentos, juntamente com seus mecanismos de ação, interações e alvos biológicos. Entretanto, o DrugBank oferece uma abordagem mais focada em farmacologia, detalhando mecanismos de ação de medicamentos, perfis de farmacocinética e farmacodinâmica, bem como interações medicamentosas. Esta orientação para o contexto clínico e farmacêutico torna o DrugBank particularmente valioso para a pesquisa de desenvolvimento de fármacos (WISHART, D. S. et al. ,2018).

No trabalho, foi escolhido o uso do repositório ChEMBL para obtenção de dados de treino e teste, devido a sua robustez de informações no que diz respeito a atividades biológicas, além de ser um serviço gratuito.

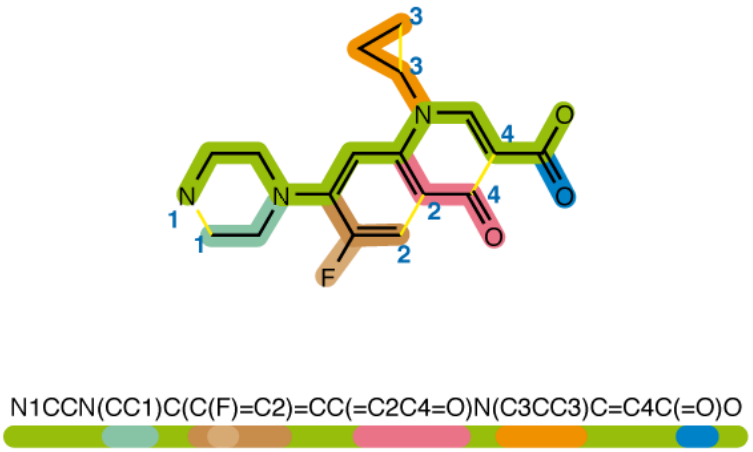
3.4. Representação Smiles

Para trabalhar com moléculas em ambiente computacional, será utilizada a chamada representação SMILES (Simplified Molecular Input Line Entry System), notação que permite descrever a estrutura de uma molécula usando uma linha de texto, simplificando a entrada e armazenamento de dados moleculares (US EPA, 2016).

Esse método de representação tem várias aplicações práticas, principalmente no campo da descoberta de fármacos. Por exemplo, a partir de bancos de dados de imagens de moléculas, é possível gerar rapidamente grandes bibliotecas de compostos em formato SMILES, que podem ser utilizadas em simulações computacionais e modelos de aprendizado de máquina para prever propriedades farmacológicas, toxicológicas e outras características relevantes (HIROHARA, M.,2018).

Além disso, a representação em SMILES facilita a integração de dados moleculares em diversas plataformas de análise, permitindo uma abordagem mais flexível e ampla na pesquisa química. O uso de SMILES também ajuda na padronização e compartilhamento de informações moleculares, aspectos críticos para a colaboração científica e o avanço da pesquisa (US EPA, 2016).

Figura 3 - Representação Smiles de molécula exemplificada



Fonte: SELLA2021-10-29T09:00:00+01:00, A. Weininger's Smiles. Disponível em: <https://www.chemistryworld.com/opinion/weiningers-smiles/4014639.article>. Acesso em: 13 dez. 2023.

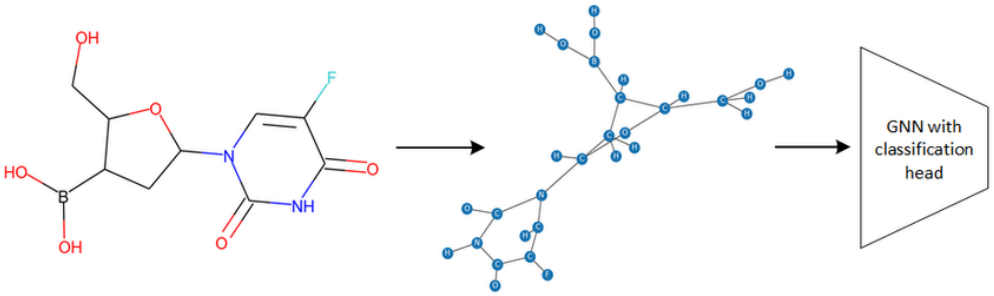
3.5. Redes Neurais Gráficas

Um dos grandes desafios históricos da implementação de técnicas de machine learning na química computacional é encontrar descritores que capturem informações de moléculas de maneira completa. Modelos mais tradicionais, como os baseados em Quantitative Structure-Activity Relationship (QSAR), frequentemente dependem da seleção manual de características que possam prever a atividade biológica ou outras propriedades químicas. Essa abordagem pode requerer um extenso conjunto de *features* moleculares, incluindo propriedades físico-químicas, descritores topológicos e geométricos, mas muitas vezes não captura a complexidade total das interações moleculares em sistemas biológicos (KWON, S. et al., 2019). Além disso, a escolha e a engenharia de características em modelos QSAR podem introduzir viéses e limitar a generalização do modelo para moléculas fora do domínio dos dados de treinamento.

Nesse contexto, as Redes Neurais Gráficas (Graph Neural Networks - GNNs) apresentam uma abordagem promissora. Este tipo de modelagem, que tem ganhado relevância consistentemente nos últimos anos, não apenas em química computacional mas como em outras aplicações (como processamento de imagens ou análises de redes sociais), possibilita a definição de suas camadas a partir de grafos (estruturas de dados que consistem em um conjunto de nós (ou vértices) e arestas que conectam pares desses nós) (SANCHEZ-LENGELING, B; 2021).

Considerando isso, a modelagem direta da estrutura de moléculas como grafos, onde átomos e ligações são tratados como nós e arestas, respectivamente, pode ser utilizada para formar a rede. Esta representação naturalmente incorpora a topologia da molécula e permite que o modelo aprenda representações de características de forma end-to-end, potencialmente capturando interações complexas sem a necessidade de engenharia de atributos pré-definida, mas também possibilitando a integração de atributos em níveis de nó e globais para aumento de sua robustez (SANCHEZ-LENGELING, B; 2021).

Figura 4 - Representação de molécula na forma de grafo



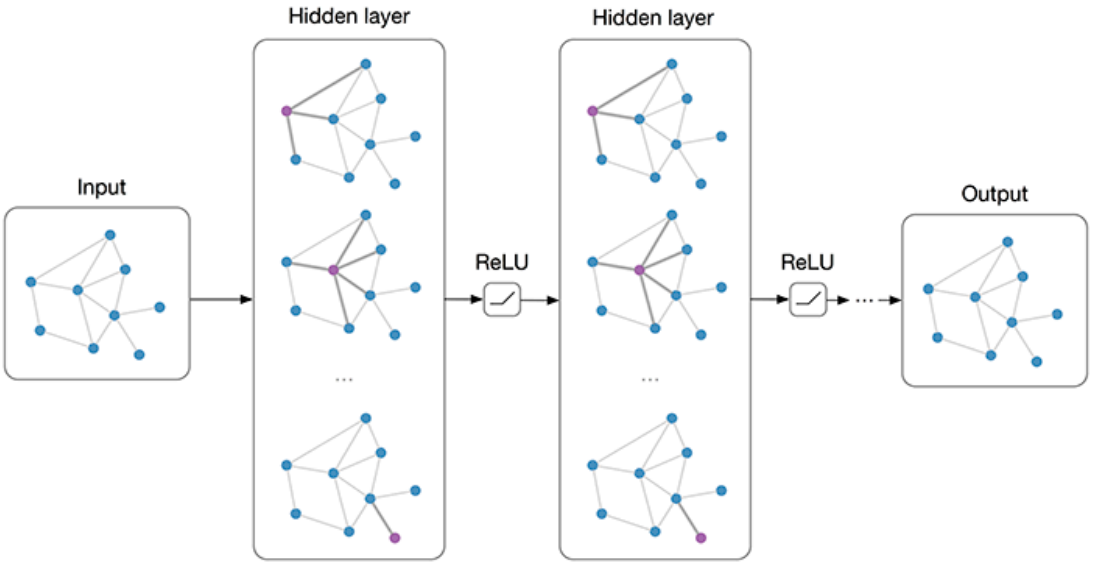
Fonte: Leng, Dawei & Guo, Jinjiang & Pan, Lurong & Li, Jie & Wang, Xinyu. (2021). Enhance Information Propagation for Graph Neural Network by Heterogeneous Aggregations.

O funcionamento das GNNs baseia-se na atualização iterativa dos estados dos nós, levando em consideração as informações dos nós vizinhos e das próprias arestas. Isso é realizado por meio de uma função de agregação, que coleta e combina informações dos vizinhos, seguida por uma função de atualização, que gera um novo estado para cada nó. Este processo permite que as GNNs capturem as relações complexas e as dependências estruturais presentes nos grafos (SANCHEZ-LENGELING, B; 2021).

A construção da rede neural para química computacional se concentra principalmente no aprendizado profundo da estrutura molecular e na interação entre átomos de maneiras que modelos tradicionais não conseguem abordar diretamente. O modelo proposto explora várias camadas de convoluções gráficas, que são fundamentais para o processamento das características dos nós e das arestas que definem as conexões entre átomos na molécula.

A figura a seguir traz uma representação de como grafos são processados em Redes Neurais Gráficas, com a representação da captura de padrões em cada uma das camadas.

Figura 5 - Representação Visual de entrada e Grafo como saída



Fonte: KARAGIANNAKOS, S. Graph Neural Networks - An overview.
Disponível em:<https://theaisummer.com/Graph_Neural_Networks/>. Acesso em: 19 de abr. 2024.

4. Metodologia

Será feita uma descrição dos passos utilizados para a construção do modelo. Todos os códigos utilizados em cada uma etapa encontram-se no apêndice.

4.1. Obtenção dos dados

Neste estudo, os dados utilizados foram extraídos do repositório ChEMBL, que contém uma vasta coleção de informações moleculares (posteriormente será feita também uma análise de quais atributos do banco de dados devem ser utilizados na construção do modelo), incluindo a notação SMILES das moléculas, considerando a seguinte estrutura:

Base “compounds” - Contém informações de propriedades gerais da molécula
Base “activities” - Informações de mecanismos biológicos e alvos.

Figura 6 - Visualização do banco de dados “compounds”

	ChEMBL ID	Name	Synonyms	Type	Max Phase	Molecular Weight	Targets	Bioactivities	AlogP	Polar Surface Area	...	Heavy Atoms	HBA (Lipinski)	HBD (Lipinski)	#RO5 Violations (Lipinski)	Molecular Weight (Monoisotopic)	Np Likeness Score	M
1	CHEMBL2236624	NaN	NaN	Small molecule	NaN	320.69	9.0	9.0	1.72	114.23	...	22.0	8.0	2.0	0.0	320.0312	-2.19	
2	CHEMBL1446444	NaN	NaN	Small molecule	NaN	389.86	16.0	18.0	3.15	79.37	...	26.0	6.0	1.0	0.0	389.0601	-2.19	
3	CHEMBL106683	NaN	NaN	Small molecule	NaN	253.37	3.0	4.0	0.87	80.39	...	14.0	4.0	3.0	0.0	252.9901	-1.35	
5	CHEMBL1398980	NaN	NaN	Small molecule	NaN	410.32	18.0	25.0	4.78	47.56	...	26.0	4.0	1.0	0.0	409.0306	-1.01	
6	CHEMBL1492189	NaN	NaN	Small molecule	NaN	479.95	5.0	8.0	4.62	86.24	...	33.0	7.0	1.0	0.0	479.0707	-2.17	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2399738	CHEMBL255056	NaN	NaN	Small molecule	NaN	593.26	7.0	10.0	10.18	49.84	...	43.0	4.0	2.0	2.0	592.3333	-0.22	
2399739	CHEMBL85912	NaN	NaN	Small molecule	NaN	499.55	1.0	8.0	3.02	71.53	...	27.0	6.0	1.0	0.0	383.1304	-0.75	
2399740	CHEMBL1594610	NaN	NaN	Small molecule	NaN	370.28	12.0	15.0	3.86	30.71	...	21.0	3.0	0.0	0.0	297.0736	-1.52	
2399741	CHEMBL3184244	NaN	NaN	Small molecule	NaN	317.29	1.0	6.0	0.77	66.48	...	12.0	3.0	4.0	0.0	167.0946	0.99	
2399742	CHEMBL3989861	DESVENLAFAXINE FUMARATE	DESVENLAFAXINE FUMARATEIDESVENLAFAXINE FUMARAT...	Small molecule	4.0	397.47	NaN	NaN	2.73	43.70	...	19.0	3.0	2.0	0.0	263.1885	0.44	

Fonte: Acervo pessoal

A seleção de dados com coerência e critérios definidos é completamente determinante para todo o seguimento do modelo, por isso, foi feita uma filtragem cuidadosa de quais dados seriam utilizados para representar a classe positiva (Moléculas que atuam como antagonista do receptor H-1) e negativa (Moléculas que não atuam como antagonistas do receptor H-1).

Para a classe positiva, foram selecionadas moléculas que tiveram atividade reportada como antagonista do receptor H1, target de ID “CHEMBL231” no repositório. Para a classe negativa, foram selecionadas moléculas que não tiveram atividade relatada com o target, mas tiveram mais de 5 bioatividades reportadas no repositório, evitando incluir moléculas que não

tenham passado por suficientes estudos biológicos. Essa metodologia garante que haja uma distinção clara entre moléculas que possuem e não possuem interação, possibilitando a detecção de padrões pelo modelo.

4.2. Preparação dos dados

A preparação dos dados é uma etapa chave no processo, e alguns passos devem ser seguidos para se adequar ao modelo e para garantir a qualidade e eficácia do trabalho desenvolvido.

4.2.1. Conversão de moléculas: Smiles para moléculas Rdkit

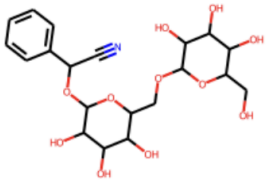
Primeiramente, para a análise computacional de moléculas, é necessário converter as moléculas SMILES em estruturas manipuláveis. A biblioteca RDKit, amplamente utilizada na química computacional, permite essa conversão seguindo um fluxo de trabalho claro e sistemático. Inicialmente, cada sequência SMILES é interpretada pelo RDKit para construir a molécula correspondente. Este processo envolve etapas de validação e normalização para assegurar que as moléculas geradas sejam quimicamente viáveis, corrigindo potenciais erros de valências ou estruturas impraticáveis.

Após a formação inicial, procede-se à adição explícita de átomos de hidrogênio, muitas vezes omitidos nas representações SMILES, para completar as valências dos átomos na molécula. Esta etapa é crucial, pois muitos estudos dependem de uma representação completa da geometria molecular para análises precisas. Além disso, as moléculas são submetidas a uma otimização geométrica que ajusta a conformação dos átomos para alcançar um estado de baixa energia potencial, condição necessária para que as estruturas sejam utilizadas em simulações e cálculos subsequentes.

Moléculas RDKit: As moléculas resultantes no RDKit são objetos sofisticados que representam estruturas químicas completas, incluindo átomos, ligações, e informações estéreo. Cada molécula pode ser manipulada, visualizada e analisada através das funcionalidades oferecidas pela biblioteca. Esses objetos são capazes de suportar uma variedade de operações químicas, como subestruturas de busca, cálculos de propriedades físico-químicas, e transformações moleculares (RDkit Documentation, 2023).

Figura 7 : Conversão de moléculas em Smiles para objetos com RDkit

```
from rdkit import Chem
from rdkit.Chem.Draw import IPythonConsole
molecule = Chem.MolFromSmiles(data[0]["smiles"])
molecule
```



Fonte:Acervo Pessoal

4.2.2. Definição de *Features* dos Grafos

Na definição de modelos de Redes Neurais Gráficas (GNN), cada nó e cada aresta podem possuir conjuntos de características que os descrevem, além de *features* globais para a molécula permitindo que o modelo processe e aprenda a partir da topologia molecular de forma detalhada. A definição das *features* utilizadas para gerar os grafos influencia diretamente a capacidade do modelo de capturar informações relevantes sobre a estrutura e as propriedades das moléculas, além de serem essenciais para a captura de padrões que levam a molécula a possuir o mecanismo (SANCHEZ-LENGELING, B. , 2021) .

Embora haja atributos específicos que tendem a ser fundamentais para o processamento de diferentes mecanismos, interações biológicos possuem alta complexidade e cada tipo de mecanismo tende a ser mais influenciado por diferentes fatores da molécula. Nesse contexto, será necessário fazer o levantamento de possíveis *features* e selecionar aqueles que melhor se relacionam com a interação como antagonista do receptor-H1. Levantamento de possíveis *features*:

***Features* por Nó - Nível Atômico**

Um nó do grafo será representado, na prática, por um vetor de (n) *features* escolhidas que descrevem um átomo específico. Algumas possibilidades de atributos a serem utilizados são:

- **Número Atômico**: Indica o tipo de átomo, determinando sua posição na tabela periódica e influenciando diretamente suas propriedades químicas fundamentais.
- **Hibridização:** Descreve o tipo de hibridação dos orbitais eletrônicos do átomo (sp, sp2, sp3), essencial para entender a geometria molecular e a natureza das ligações químicas.
- **Eletronegatividade:** Medida da tendência de um átomo em atrair elétrons de uma ligação química, crucial para prever a polaridade e a reatividade das moléculas.
- **Raio Atômico**: Reflete o tamanho do átomo, o que afeta como os átomos se aproximam uns dos outros e formam estruturas moleculares.
- **Estado de Oxidação:** Indica o grau de ganho ou perda de elétrons de um átomo em comparação com seu estado neutro, influenciando a formação de diferentes tipos de ligações e compostos químicos.
- **Massa Atômica:** Representa a massa de um átomo, contribuindo para o cálculo da massa molar da molécula e afetando propriedades como densidade e peso molecular.

***Features* Globais - Nível Molecular**

Atributos globais, referentes a molécula como um todo, são incorporados no grafo molecular como características adicionais que são processadas juntamente com as características dos nós nas camadas subsequentes da rede, permitindo que o modelo capture não apenas informações estruturais locais, mas também contextos globais que são essenciais para previsões precisas de propriedades e atividades moleculares. Algumas propriedades globais relevantes são:

- **Molecular Weight (Peso Molecular):** Reflete a massa total da molécula e é fundamental para entender a distribuição, metabolismo e excreção da molécula no organismo.
- **AlogP**: Uma medida da lipofilicidade da molécula, indicativa de sua solubilidade em lipídios e importante para a permeabilidade celular.
- **Violações das 5 Regras de Lipinski:** Fornecem uma heurística para avaliar a 'drogabilidade' de compostos, sugerindo se uma molécula tem propriedades que seriam favoráveis em um medicamento oralmente ativo, onde estabelece que uma molécula para ser um bom fármaco deve apresentar valores para 4 parâmetros múltiplos de 5:

log P maior ou igual a 5, Massa Molecular menor ou igual a 500, aceptores de ligação de Hidrogênio menor ou igual a 10 e doadores de ligação de hidrogênio menor ou igual a 5.

- **Número de Anéis Aromáticos:** Total de anéis aromáticos presentes na molécula
- **Eficiência do Ligante BEI:** Relaciona a eficácia do ligante com seu tamanho, fornecendo uma medida da eficiência de uma molécula como ligante.
- **Área de Superfície Polar (TPSA):** medida da soma das superfícies de todos os átomos polares (normalmente oxigênio, nitrogênio e átomos associados a hidrogênios) em uma molécula.

A definição de features a serem utilizadas para a construção de modelos de Redes Neurais Gráficas possui nuances que a torna significativamente mais complexa que em modelos de machine learning tradicionais. Enquanto modelos tradicionais como árvores de decisão oferecem uma clara interpretabilidade das *features*, em GNNs, a contribuição de cada feature é influenciada pelas características dos nós vizinhos e pela estrutura do grafo, tornando a interpretação dos resultados mais complexa. (ZHOU, J. , 2021). Nesse contexto, serão definidas as *features* a partir de permutações de consideração, avaliando a perda para o conjunto de teste e a performance do modelo.

4.2.3. Normalização de Atributos Globais

A normalização dos atributos globais é uma etapa fundamental no pré-processamento de dados para modelos de aprendizado de máquina, incluindo Redes Neurais Gráficas (GNNs). Nesta etapa, ajusta-se a escala das variáveis numéricas do conjunto de dados para garantir que elas contribuam de maneira equitativa para o processo de aprendizagem do modelo, minimizando o risco de que variáveis com maiores ordens de magnitude dominem o treinamento.

Na construção do modelo, foi utilizada a Z-score Normalization para todos os atributos numéricos considerados. Neste método, os dados são escalados de modo que tenham uma média de zero e um desvio padrão de um. Isso é alcançado subtraindo a média dos dados e dividindo pelo desvio padrão, como indicado:

$$z = \frac{x-\mu}{\sigma} \qquad (1)$$

Sendo z a variável normalizada, μ a média da amostra e σ o desvio padrão amostral
Esta abordagem é benéfica quando os atributos apresentam uma distribuição aproximadamente normal e é eficaz para reduzir o impacto de *outliers*.

4.2.4. Criação de Data Loaders

Após a formulação de grafos e normalização de variáveis de atributos numéricos, o último passo antes de utilizar o modelo é a criação de Data Loaders. Data loaders são ferramentas que gerenciam o carregamento e a preparação dos dados para serem alimentados aos modelos de forma eficiente, ao facilitar o processo de iterar sobre os dados em lotes, o que é essencial para treinar modelos em grandes conjuntos de dados que não caberiam na memória se carregados todos de uma vez (Stanford University, 2022).

Para isso, inicialmente, foi realizada a construção de uma lista de objetos de dados, onde cada objeto representa uma molécula e suas características correspondentes. A partir de um dataframe pré-definido, que já continha todas as informações necessárias, iterou-se sobre cada linha para processar e transformar essas informações em grafos compatíveis com o framework PyTorch Geometric.

4.3. Construção da Rede Neural Gráfica

Para a aplicação das redes neurais gráficas, será utilizada a biblioteca PyTorch Geometric, biblioteca de extensão para PyTorch dedicada a Deep Learning em grafos e outras estruturas de dados irregulares que simplifica a implementação de GNNs, fornecendo várias funções e classes prontas para uso. O código utilizado encontra-se no Apêndice C.

4.3.1. Definição de Camadas Convolucionais

As camadas de convolução para grafos, ou GCNConv, são projetadas especificamente para trabalhar com dados estruturados como grafos. Essas camadas aplicam filtros de convolução sobre os nós do grafo, considerando suas conexões locais (arestas) para atualizar suas características. Isso é similar à convolução em redes neurais convencionais usadas para processamento de imagens, mas adaptado para dados não euclidianos. Cada nó no grafo recebe informações de seus vizinhos através do processo de agregação, onde a função de agregação pode ser uma simples média, uma soma ou um máximo das características dos

vizinhos, permitindo que o modelo capture tanto informações locais quanto globais dentro do grafo (SANCHEZ-LENGELING, B , 2021).

Paralelamente, o modelo utiliza a função de ativação ReLU (Rectified Linear Unit) nas camadas convolucionais. A ReLU é escolhida por suas propriedades que ajudam a acelerar a convergência do treinamento da rede neural, ao mesmo tempo que introduz não-linearidade ao processo, permitindo que o modelo aprenda relações complexas nos dados. A função ReLU é definida como $f(x)=max(0,x)$, onde x é a entrada para o neurônio. Essa função tem a vantagem de ser menos suscetível ao problema do desvanecimento do gradiente, que é comum em redes profundas, pois mantém a derivada em um valor constante (1) para todas as entradas positivas, garantindo uma atualização eficaz dos pesos durante o treinamento (SANCHEZ-LENGELING, B. , 2021). Essa combinação de GCNConv e ReLU permite um tratamento eficaz e eficiente das características dos grafos, resultando em um aprendizado mais robusto e representativo das estruturas moleculares.

4.3.2. Consideração de Pesos - Desbalanço de classes

Devido ao desbalanceamento inerente dos dados - um problema comum em dados de química computacional onde algumas classes de respostas são muito menos representadas do que outras -, uma técnica de compensação de pesos é aplicada. Isso envolve ajustar a função de perda para penalizar mais fortemente os erros em classes sub-representadas, equilibrando assim a influência de cada classe no modelo (SINGH, K, 2023). Essa abordagem é crucial para assegurar que o modelo não favoreça a maioria, ignorando as características importantes das moléculas menos comuns, mas potencialmente mais interessantes.

A inicialização dos pesos é feita na linha de código self.apply(init_weights), usando o método de Kaiming He, ideal para camadas acompanhadas pela função de ativação ReLU, para assegurar que as ativações inicialmente não propaguem valores extremos que podem levar a gradientes explosivos ou desaparecidos. O modelo é otimizado usando o algoritmo Adam, um método de otimização baseado em gradiente que ajusta a taxa de aprendizagem de cada parâmetro individualmente. Isso permite uma convergência mais rápida e eficiente em treinamentos profundos.

4.3.3. Prevenção de Overfitting - Camadas Densas e Dropout

Seguindo a camada de *pooling*, as características são passadas por camadas densas lineares, que podem aprender combinações não-lineares das características. O *dropout* é inserido entre as camadas densas para mitigar o risco de *overfitting*, uma preocupação comum em modelos de *deep learning*, especialmente em conjuntos de dados complexos e variados como os encontrados na química. O *dropout* funciona "desligando" aleatoriamente uma porcentagem dos neurônios durante a fase de treinamento, forçando a rede a aprender caminhos redundantes, aumentando assim a robustez do modelo (SRIVASTAVA, N. , 2014).

4.4. Treinamento do modelo

No processo de treinamento do modelo, a partir do conjunto de treino, foi utilizado o otimizador Adam. Adam, uma abreviação para "Adaptive Moment Estimation", é um algoritmo de otimização que ajusta as taxas de aprendizado de cada parâmetro do modelo. Este otimizador é preferido por sua eficiência computacional e por requerer pouca memória, além de ser bem adequado para dados e parâmetros grandes devido à sua capacidade de combinar os benefícios dos otimizadores AdaGrad e RMSProp. (BROWNLEE, J. , 2021) Ao utilizar o Adam, espera-se que a taxa de perda diminua de forma eficiente e estável, evitando os problemas comuns de taxas de aprendizado fixas que podem resultar tanto em convergência lenta quanto em oscilações indesejadas na perda durante o treinamento.

As métricas de treinamento incluem a perda calculada pela função de custo, neste caso, a entropia cruzada binária com logits (BCEWithLogitsLoss). Esta função é particularmente adequada para problemas de classificação binária e é utilizada para medir o desempenho do modelo ao comparar as saídas previstas com os rótulos verdadeiros (TA, V. et al. , 2021). A perda representa a discrepância entre as probabilidades previstas e os valores reais, onde uma menor perda indica um modelo melhor. Durante o treinamento, a redução contínua da perda é um indicativo de aprendizagem eficaz, enquanto a estagnação ou aumento pode sinalizar problemas como o *overfitting* ou sub ajuste.

4.5. Teste e Avaliação dos Resultados

A última etapa do trabalho é a avaliação da fase de teste do modelo treinado com dados não observados durante o treinamento, fundamental para verificar sua capacidade de generalização. Diversas métricas serão empregadas para medir a eficácia do modelo,

considerando as diferentes nuances que ele possui. Ao lidar com classes desbalanceadas, como é o caso, não faz sentido utilizar como métrica a acurácia geral do modelo. Uma exemplificação clara para constatar o fato é imaginar que o modelo previsse 100% das moléculas como não ativas: Obteríamos mais de 95% de acurácia, mas isso jamais seria um indicativo de boa performance do modelo.

Entre elas, estão a matriz de confusão, que destaca-se por oferecer uma análise detalhada do desempenho do modelo em cada classe, identificando verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos. Essa ferramenta permite um exame mais minucioso do comportamento do modelo, com foco especial na sensibilidade (*recall*) e precisão, aspectos vitais para compreender suas limitações e eficiência.

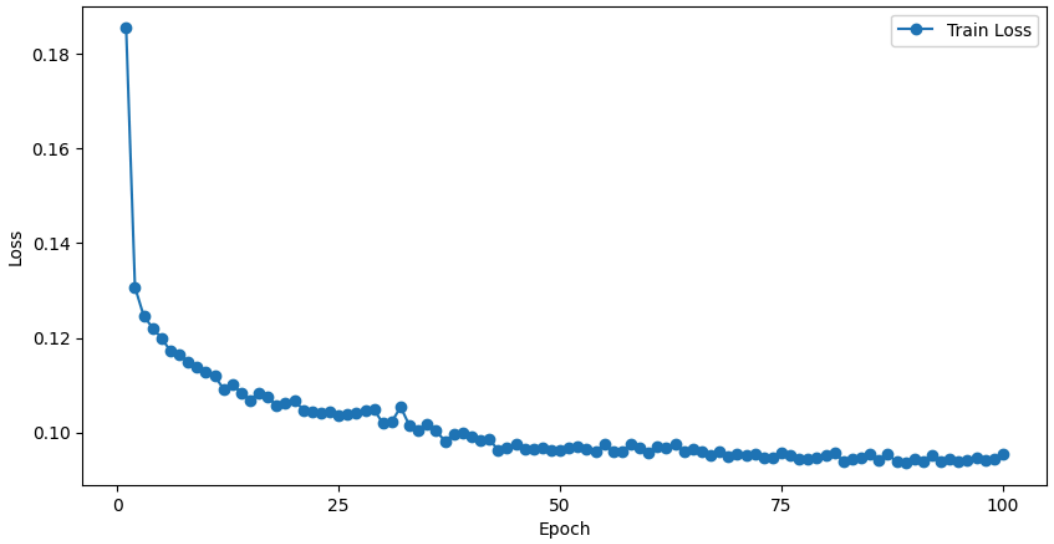
Além da matriz de confusão, é também construída uma curva ROC, que ilustra a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos em vários limiares de decisão. A área sob a curva (AUC) representa uma medida agregada do desempenho do modelo em todas as classes de limiares, fornecendo um indicador único e abrangente da eficácia do modelo na classificação das moléculas de acordo com a sua interação com o receptor de interesse. Quanto mais próxima a curva estiver do canto superior esquerdo do gráfico, maior será a capacidade do modelo de oferecer previsões precisas com um menor número de falsos positivos, essencial para a confiança na seleção de potenciais compostos terapêuticos (NARKHEDE, S , 2018).

5. Resultados

Visando a uma melhor capacidade classificatória do modelo, foram incorporadas aos grafos *features* globais da molécula. A combinação de *features* que demonstrou a melhor performance de treinamento considerou 5 fatores: AlogP, número de anéis aromáticos, Violações das 5 regras de Lipski, peso molecular e área de superfície polar total. Os resultados obtidos para o caso foram:

5.1 Treinamento e otimização:

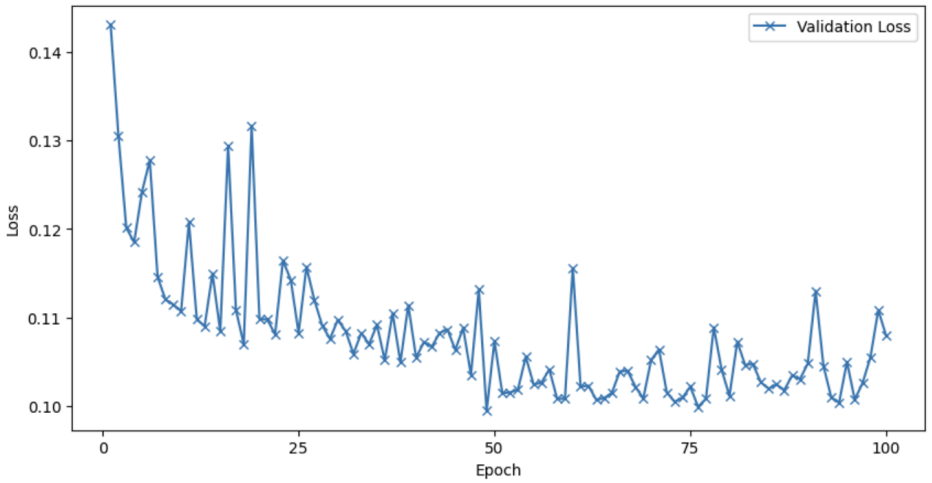
Figura 8 - Gráfico de Perdas por Época de treinamento



Observando o gráfico de perdas obtido, é possível notar que inicialmente há uma queda acentuada na perda, indicando que o modelo está aprendendo rapidamente a partir dos dados, seguido por uma estabilização a partir da época 50. O gráfico apresenta um comportamento esperado de um modelo que está aprendendo e melhorando sua capacidade de previsão ao longo do tempo.

Para uma análise mais aprofundada da possibilidade de overfitting, foi plotado o gráfico de perdas em função de épocas para o conjunto de teste. Em casos de *overfitting*, enquanto houvesse diminuição da perda para o treino, haveria aumento da perda para o teste.

Figura 9 - Gráfico de perdas por Época - Conjunto de teste

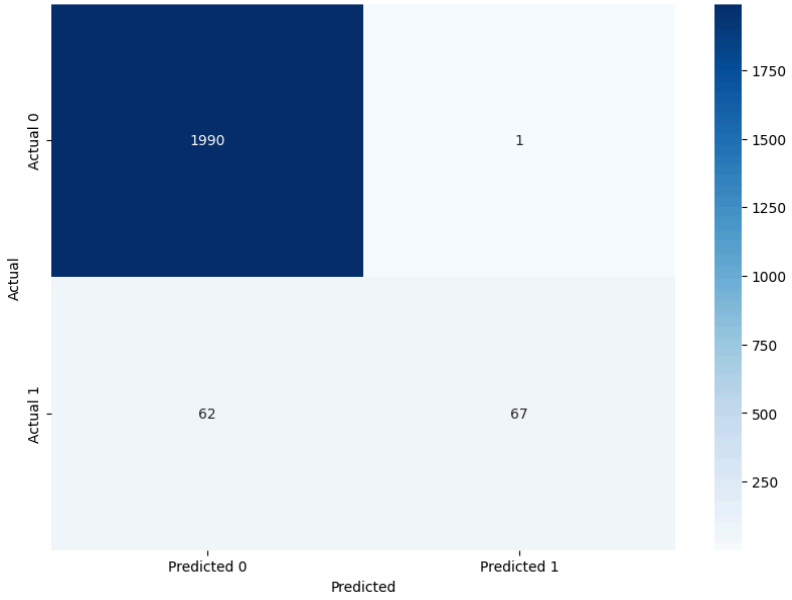


Observando o gráfico de perdas para o conjunto de teste, podemos observar que, embora hajam mais oscilações do que no conjunto de treino - o que é justificável pela natureza estocástica do otimizador, que acaba sendo menos compensada em um menor conjunto de dados - a perda também segue a tendência de queda inicial seguida de estabilização, sem tendência de aumento observada.

5.2 Matriz de Confusão

Após o treinamento do modelo, pode-se construir a matriz de confusão para uma interpretação mais profunda da performance da classificação para o conjunto de teste.

Figura 10 - Matriz de Confusão

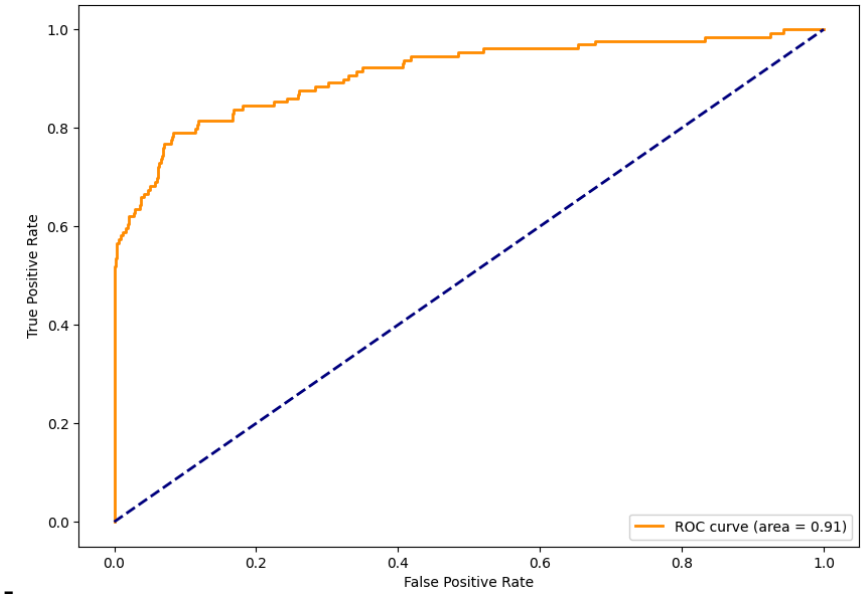


É possível notar que, de 139 moléculas com atividade como antagonistas do receptor h-1 no conjunto de teste, 67 foram detectadas detectadas como negativas. Um ponto a se notar na análise da performance do modelo é a baixa taxa de falsos positivos (substâncias que não possuem atividade classificadas como se tivessem), fator importantíssimo na possível implementação do modelo como suporte no desenvolvimento de fármacos.

5.3 Curva ROC

Para uma visualização completa da capacidade de discriminação do modelo, é construida a curva ROC:

Figura 11 - Curva ROC para modelo com *features* globais



-

A partir da curva obtida, podemos fazer algumas observações:

Área sob a curva (AUC): O valor da área sob a curva ROC é de 0.91, o que indica um alto grau de capacidade de discriminação do modelo, indicando que o modelo tem uma probabilidade significativa de diferenciar corretamente entre as moléculas que interagem e as que não interagem com o receptor alvo.

Além disso, podemos notar que a taxa de verdadeiro positivo é alta na maior parte da curva, o que sugere que o modelo tem sucesso na identificação da maioria das moléculas ativas (interações verdadeiras com o receptor). Isso é crucial em aplicações farmacológicas onde não se quer perder potenciais candidatos a medicamentos. O modelo também apresenta uma taxa de falso positivo moderadamente baixa, o que é desejável.

6. Conclusões

Neste trabalho, foi explorado o panorama global da aplicação da inteligência artificial no desenvolvimento de fármacos e a construção de um modelo que aplica o conhecimento para auxiliar o desenvolvimento de drogas anti-histamínicas. Primeiramente, foram abordados os principais trabalhos da literatura que atuaram nas áreas de design molecular, identificação e validação de alvos biológicos, além de Testes Pré-Clínicos e Toxicologia. Além disso, foi feita uma pesquisa de empresas que já utilizam dos conhecimentos para aplicações práticas no mercado atualmente, citando algumas de suas atuações.

Na construção do modelo para prever a interação de moléculas como antagonistas do receptor anti-histamínicos utilizando redes neurais gráficas, foram atingidos resultados significativos na capacidade da previsão do modelo. Com os resultados, pode-se concluir que, embora existam oportunidades para otimização futura, a abordagem adotada já é capaz de servir como uma ferramenta auxiliar valiosa no processo de descoberta e desenvolvimento de novos fármacos anti-histamínicos. O alto desempenho do modelo enfatiza sua utilidade potencial na redução de custos e tempos associados às fases iniciais de pesquisa farmacêutica.

Por fim, pode-se concluir que o uso de inteligência artificial no desenvolvimento de fármacos é um campo extremamente promissor e com um potencial de revolução na farmacologia imensurável. Pode-se observar que o conceito já é usado em diversas aplicações práticas, mas ainda apresenta desafios notáveis e potenciais de otimização.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] MOCK, M. et al. AI can help to speed up drug discovery — but only if we give it the right data. Nature, v. 621, n. 7979, p. 467–470, 1 set. 2023.

[2] CHITHRANANDA, S.; GRAND, G.; RAMSUNDAR, B. ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. arXiv:2010.09885 [physics, q-bio], 23 out. 2020.

[3] JIN, W. et al. Predicting Organic Reaction Outcomes with Weisfeiler-Lehman Network. [s.l: s.n.]. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2017/file/ced556cd9f9c0c8315cfbe0744a3baf0-Paper.pdf>. Acesso em: 13 dez. 2023.

[4] D. White, Andrew. Graph Neural Networks — deep learning for molecules & materials. Disponível em: <https://dmol.pub/dl/gnn.html>. Acesso em: 13 dez. 2023.

[5] Leng, Dawei & Guo, Jinjiang & Pan, Lurong & Li, Jie & Wang, Xinyu. (2021). Enhance Information Propagation for Graph Neural Network by Heterogeneous Aggregations.

[6] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, Vijay Pande, MoleculeNet: A Benchmark for Molecular Machine Learning, arXiv preprint, arXiv: 1703.00564, 2017.

[7] CRIADO, P. R. et al. Histamina, receptores de histamina e anti-histamínicos: novos conceitos. Anais Brasileiros de Dermatologia, v. 85, n. 2, p. 195–210, abr. 2010.

[8] Öztürk, H., Özgür, A., & Ozkirimli, E. (2018). DeepDTA: deep drug–target binding affinity prediction. Bioinformatics, 34(17), i821-i829.

[9] Mayr, A., Klambauer, G., Unterthiner, T., & Hochreiter, S. (2016). DeepTox: Toxicity Prediction using Deep Learning. Frontiers in Environmental Science, 3, 80.

[10] OLIVECRONA, M. et al. Molecular de-novo design through deep reinforcement learning. Journal of Cheminformatics, v. 9, n. 1, 4 set. 2017.

[11] ZHOU, J. et al. Graph Neural Networks: A Review of Methods and Applications. Disponível em: <https://arxiv.org/abs/1812.08434>.

[12] SANCHEZ-LENGELING, B. et al. A Gentle Introduction to Graph Neural Networks. Distill, v. 6, n. 8, 17 ago. 2021.

[13] CLEVELAND CLINIC. Antihistamines: Definition, Types & Side Effects. Disponível em: <https://my.clevelandclinic.org/health/drugs/21223-antihistamines>.

[14] PAUFERRO, M. R. V. Desenvolvimento de medicamentos: desde a pesquisa até a prateleira.Disponível em: <https://nexxto.com/desenvolvimento-medicamentos-desde-a-pesquisa-ate-a-prateleira/#:~:text=O%20desenvolvimento%20de%20um%20novo>.

[15] SMILES Tutorial | Research | US EPA. Disponível em: <https://archive.epa.gov/med/med_archive_03/web/html/smiles.html>.

[16] A detailed example of data loaders with PyTorch. Disponível em: <https://stanford.edu/~shervine/blog/pytorch-how-to-generate-data-parallel>.

[17] JASON BROWNLEE. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. Disponível em: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>.

[18] NARKHEDE, S. Understanding AUC - ROC Curve. Disponível em: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>.

[19] KIRKPATRICK, P. Artificial intelligence makes a splash in small-molecule drug discovery. Biopharma Dealmakers, 17 maio 2022.

[20] WERNER, M. When People With Glaucoma Should Avoid Allergy And Decongestant Medications - Glaucoma Research Foundation. Disponível em: <https://glaucoma.org/articles/when-people-with-glaucoma-should-avoid-allergy-and-decongestant-medications>. Acesso em: 24 abr. 2024.

[21] HIROHARA, M. et al. Convolutional neural network based on SMILES representation of compounds for detecting chemical motif. BMC Bioinformatics, v. 19, n. S19, dez. 2018.

[22] Getting Started with the RDKit in Python — The RDKit 2023.03.1 documentation. Disponível em: <https://www.rdkit.org/docs/GettingStartedInPython.html>.

[23] SINGH, K. How to Improve Class Imbalance using Class Weights in Machine Learning. Disponível em: <https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/>.

[24] MARTIN, G. This company uses AI to accelerate drug discovery. Disponível em: <https://www.oreilly.com/content/this-company-uses-ai-to-accelerate-drug-discovery/>. Acesso em: 20 abr. 2024.

[25] UK Research and Innovation. Exscientia: a clinical pipeline for AI-designed drug candidates. Disponível em: <https://www.ukri.org/who-we-are/how-we-are-doing/research-outcomes-and-impact/bbsrc/exscientia-a-clinical-pipeline-for-ai-designed-drug-candidates/>

[26] BUNTZ, B. BenevolentAI exploring novel AI-driven drug discovery methods. Disponível em: <https://www.drugdiscoverytrends.com/benevolentai-is-pioneering-ai-driven-drug-discovery-methods/>.

[27] SRIVASTAVA, N. et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, v. 15, p. 1929–1958, 2014.

[28] WISHART, D. S. et al. DrugBank 5.0: a major update to the DrugBank database for 2018. Nucleic acids research, v. 46, n. D1, p. D1074–D1082, 2018.

[29] GAULTON, A. et al. The ChEMBL database in 2017. Nucleic Acids Research, v. 45, n. D1, p. D945–D954, 28 nov. 2016.

[30] KIM, S. et al. PubChem Substance and Compound databases. Nucleic Acids Research, v. 44, n. D1, p. D1202–D1213, 22 set. 2015.

[31] TA, V. et al. Binary Classification: An Introductory Machine Learning Tutorial for Social Scientists. Journal of Methods and Measurement in the Social Sciences, Vol. 12, No. 2, 56-94, 2021

[32] KWON, S. et al. Comprehensive ensemble in QSAR prediction for drug discovery. BMC Bioinformatics, v. 20, n. 1, 26 out. 2019.

APÊNDICE

APÊNDICE A - Código: Importação de bibliotecas utilizadas

```
# Pacotes gerais
import numpy as np
import pandas as pd
import tensorflow as tf
import warnings

# Módulos RDKit - Processamento de moléculas

from rdkit import Chem
from rdkit import RDLogger
from rdkit.Chem.Draw import IPythonConsole
from rdkit.Chem.Draw import MolsToGridImage
from rdkit.Chem import AllChem

# Módulos Torch - Modelagem Geral
import torch
from torch_geometric.data import Data
from torch_geometric.data import DataLoader
import torch.nn.functional as F
from torch_geometric.nn import GCNConv, global_mean_pool,
global_max_pool
from torch.nn import Linear

#Timeout para execução de funções
from concurrent.futures import ThreadPoolExecutor, TimeoutError

#Análise de resultados
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, roc_auc_score, classification_report,
confusion_matrix, roc_curve, auc
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns
```

APÊNDICE B - Código: Conversão de smiles em moléculas Rdkit

```
#Converter moleculas para mol_rdkit
def smiles_to_mol(smiles):
   try:
     mol=Chem.MolFromSmiles(smiles)
     if mol is None:  # Verificar se a molécula foi criada
         return None

     mol = Chem.AddHs(mol)
      if AllChem.EmbedMolecule(mol, AllChem.ETKDG()) < 0:  # Verificar
se a molécula foi incorporada com sucesso
         return None
     return mol
   except:
     return None
array=[]
count=0

def process_smiles_and_save(df, batch_size=10000):
   array = []
   for index, smiles in enumerate(df['Smiles']):
       if (index) % 100 == 0:
         print(index)

       mol = smiles_to_mol(smiles)
       array.append(mol)
       if (index + 1) % batch_size == 0:
           df_partial = df.iloc[:index + 1].copy()
           df_partial['rdkit_mol'] = array
           df_partial.to_csv(AUX_PATH+'chemBlmolecules_rdkit.csv')
           print(f'Saved {index + 1} records.')

   if len(df) % batch_size != 0:
       df['rdkit_mol'] = array
       df.to_csv(AUX_PATH+'chemBlmolecules_rdkit_final.csv')
       print(f'Saved final batch of records.')

   return(array)
```

APÊNDICE C - Código: Conversão moléculas Rdkit em grafos

```
def mol_to_graph(mol):
   if mol is None:
     return None

   atoms = [atom.GetAtomicNum() for atom in mol.GetAtoms()]
   edge_index = []
   edge_attr = []

   for bond in mol.GetBonds():
       start, end = bond.GetBeginAtomIdx(), bond.GetEndAtomIdx()
       edge_index.append((start, end))
       edge_index.append((end, start))

   node_features = torch.tensor(atoms, dtype=torch.long).unsqueeze(-1)
                       edge_index       =       torch.tensor(edge_index,
dtype=torch.long).t().contiguous()

   return Data(x=node_features, edge_index=edge_index)
```

APÊNDICE D - Código: Definição de Rede Neural

```
def init_weights(m):
   if isinstance(m, nn.Linear):
                    nn.init.kaiming_normal_(m.weight,   mode='fan_out',
nonlinearity='relu')

class GNN(torch.nn.Module):
        def   __init__(self,   num_node_features,   num_global_features,
dropout_rate=0.5):
       super(GNN, self).__init__()
       self.apply(init_weights)
       self.conv1 = GCNConv(num_node_features, 16)
       self.conv2 = GCNConv(16, 32)
       self.fc_global = torch.nn.Linear(num_global_features, 16)
       self.fc = torch.nn.Linear(32 + 16, 1)
       self.dropout = nn.Dropout(dropout_rate)  # módulo de Dropout

   def forward(self, data, global_features):
       x, edge_index, batch = data.x, data.edge_index, data.batch

       x = F.relu(self.conv1(x, edge_index))
            x  =  self.dropout(x)    #  dropout  após  a  primeira  camada
convolucional
       x = F.relu(self.conv2(x, edge_index))
       x = global_mean_pool(x, batch)  # Pooling global

       global_features = F.relu(self.fc_global(global_features))
            global_features  =  self.dropout(global_features)    #  Aplica
dropout nas características globais

       x = torch.cat((x, global_features), dim=1)  # Concatenação
       x = self.fc(x)  # Saída final
       return x

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = GNN(num_node_features=2, num_global_features=5)
model.to(device)
```

APÊNDICE E - Código: Criação de Data Loaders

```
# Suponha que df_compounds já esteja definido e contenha as colunas necessárias
graph_data_list = []
for _, row in df_compounds_100.iterrows():
    graph_data = row['graph']  # Aqui assumimos que 'graph' já é um objeto Data contendo x e edge_index
   if graph_data is not None:
       # Define o rótulo
            graph_data.y = torch.tensor([row['histamine_antagonist']], dtype=torch.long)

       # Adiciona características globais

            global_features = torch.tensor([row['Aromatic Rings n'], row['Polar Surface Area n'], row['#RO5 Violations n'], row['Molecular Weight n'], row['AlogP n']], dtype=torch.float).unsqueeze(0)

       graph_data.global_features = global_features
       graph_data_list.append(graph_data)

# Filtrar quaisquer elementos None
graph_data_list = [data for data in graph_data_list if data is not None and data.y is not None]
```

APÊNDICE F - Treinamento e Teste

```
def train(model, loader, optimizer, criterion, device):
    model.train()
    total_loss = 0
    losses = []  # Lista para armazenar a perda de cada época

    for data in loader:
        data.to(device)
        global_features = data.global_features.to(device)

        optimizer.zero_grad()
        output = model(data, global_features).squeeze()
         output = torch.clamp(output, min=-100, max=100)  # Clamping para
evitar valores extremos

        if torch.isnan(output).any():
            print("Nan found in output")
            continue

        loss = criterion(output, data.y.float())

        if torch.isnan(loss):
            print("Nan in loss")
            continue

        loss.backward()
        optimizer.step()
        total_loss += loss.item() if not torch.isnan(loss).item() else 0

    average_loss = total_loss / len(loader)
    losses.append(average_loss)  # Armazenar a perda média da época
    return losses

# Configuração do modelo, otimizador e critério
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model.to(device)
optimizer = torch.optim.Adam(model.parameters(), lr=0.01)
criterion = torch.nn.BCEWithLogitsLoss()
```

```
# Treinamento e plotagem das perdas
epoch_losses = []  # Lista para armazenar as perdas de todas as épocas
for epoch in range(100):  # Suponha que você treine por 10 épocas
      epoch_loss  =  train(model,  train_loader,  optimizer,  criterion,
device)
   epoch_losses.extend(epoch_loss)  # Adicione as perdas desta época
   print(f'Epoch {epoch}, Train Loss: {epoch_loss[-1]}')
```