UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Modelagem QSAR (Relação Quantitativa Estrutura-Atividade), busca por similaridade e triagem virtual para a identificação de inibidores de Acetilcolinesterase (AChE) para a doença de Alzheimer

Leandro Pedrosa

Monografia - MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

Leandro Pedrosa

Modelagem QSAR (Relação Quantitativa Estrutura-Atividade), busca por similaridade e triagem virtual para a identificação de inibidores de Acetilcolinesterase (AChE) para a doença de Alzheimer

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientadora: Profa. Dra. Tatiane Nogueira Rios

Versão original

São Carlos

2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi, ICMC/USP, com os dados fornecidos pelo(a) autor(a)

Pedrosa, Leandro
S856m Modelagem QSAR (Relação Quantitativa Estrutura-Atividade), busca por similaridade e triagem virtual para a identificação de inibidores de Acetilcolinesterase (AChE) para a doença de Alzheimer / Leandro Pedrosa ; orientadora Tatiane Nogueira Rios. – São Carlos, 2023.
118 p. : il. (algumas color.) ; 30 cm.

Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2023.

1. Relação Quantitativa Estrutura-Atividade. 2. Aprendizado de Máquina. 3. Doença de Alzheimer. I. Rios, Tatiane Nogueira, orient. II. Título.

Leandro Pedrosa

Modelagem QSAR (Relação Quantitativa Estrutura-Atividade), busca por similaridade e triagem virtual para a identificação de inibidores de Acetilcolinesterase (AChE) para a doença de Alzheimer

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Advisor: Profa. Dra. Tatiane Nogueira Rios

Original version

São Carlos

2023

RESUMO

Pedrosa, L. **Modelagem QSAR (Relação Quantitativa Estrutura-Atividade), busca por similaridade e triagem virtual para a identificação de inibidores de Acetilcolinesterase (AChE) para a doença de Alzheimer**. 2023. 118p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

A doença de Alzheimer representa um desafio considerável tanto no campo da Inteligência Artificial (IA) quanto na pesquisa em ciências da saúde. Nesse cenário desafiador, esta pesquisa foi direcionada ao desenvolvimento de abordagens terapêuticas inovadoras para combater essa doença neurodegenerativa. Entre essas abordagens, destaca-se a aplicação da Inteligência Artificial, em particular a modelagem QSAR (Relação Quantitativa Estrutura-Atividade), combinada com técnicas de aprendizado de máquina (*machine learning*) e aprendizado profundo (*deep learning*). A enzima acetilcolinesterase (AChE) desempenha um papel crucial na degradação da acetilcolina no cérebro, afetando diretamente a função cognitiva. Inibir a AChE pode levar à acumulação de acetilcolina, o que, por sua vez, pode melhorar a transmissão neural e aliviar os sintomas da doença de Alzheimer. Neste estudo, vários modelos QSAR foram desenvolvidos utilizando técnicas de IA, como SVM (Máquina de Vetores Suporte), *Random Forest*, *Multilayer Perceptron* e *TensorFlow Keras*. Além disso, foram usados descritores moleculares para capturar as características específicas dos compostos químicos, como *Fingerprints* de Morgan, SiRMS (*Simplex Representation of Molecular Structure*) e RDKit. Esses modelos foram treinados e avaliados por meio de validação cruzada estratificada, utilizando métricas estatísticas para determinar a sua eficácia. Os modelos mais promissores, com base em seus hiperparâmetros e desempenho na validação cruzada, foram selecionados para uma etapa adicional de triagem virtual. Essa etapa envolveu a busca por compostos quimicamente semelhantes aos candidatos iniciais, a fim de identificar novos inibidores da enzima AChE. Essa abordagem de modelagem e triagem virtual, que combina resultados de modelos e busca por similaridade, tem o potencial de contribuir significativamente para a descoberta de novos compostos promissores no tratamento e prevenção da doença de Alzheimer. A integração de técnicas de IA, modelagem molecular e triagem virtual oferece uma estratégia inovadora para abordar os desafios associados à doença de Alzheimer, e os resultados deste estudo têm o potencial de impactar positivamente o desenvolvimento de terapias para essa condição debilitante.

Palavras-chave: Relação Quantitativa Estrutura-Atividade. Classificação Binária. Busca por Similaridade. Triagem Virtual. Aprendizado de Máquina. Aprendizado Profundo de Máquina. Doença de Alzheimer.

ABSTRACT

Pedrosa, L. **QSAR (Quantitative Structure-Activity Relationship) modeling, similarity search, and virtual screening for identifying Acetylcholinesterase (AChE) inhibitors for Alzheimer's disease**. 2023. 118p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

Alzheimer's disease represents a considerable challenge in both the field of Artificial Intelligence (AI) and health sciences research. In this challenging scenario, this research was aimed at developing innovative therapeutic approaches to combat this neurodegenerative disease. Among these approaches, the application of Artificial Intelligence stands out, in particular QSAR (Quantitative Structure-Activity Relationship) modeling, combined with machine learning and deep learning techniques. The enzyme acetylcholinesterase (AChE) plays a crucial role in the breakdown of acetylcholine in the brain, directly affecting cognitive function. Inhibiting AChE can lead to the accumulation of acetylcholine, which in turn can improve neural transmission and alleviate the symptoms of Alzheimer's disease. In this study, several QSAR models were developed using AI techniques such as SVM, Random Forest, Multilayer Perceptron and TensorFlow Keras. Furthermore, molecular descriptors were used to capture the specific characteristics of chemical compounds, such as Morgan Fingerprints, SiRMS and RDKit. These models were trained and evaluated through stratified cross-validation, using statistical metrics to determine their effectiveness. The most promising models, based on their hyperparameters and cross-validation performance, were selected for an additional virtual screening step. This step involved the search for compounds chemically similar to the initial candidates, in order to identify new inhibitors of the AChE enzyme. This virtual modeling and screening approach, which combines model outputs and similarity searching, has the potential to contribute significantly to the discovery of promising new compounds in the treatment and prevention of Alzheimer's disease. The integration of AI techniques, molecular modeling and virtual screening offers an innovative strategy for addressing the challenges associated with Alzheimer's disease, and the results of this study have the potential to positively impact the development of therapies for this debilitating condition.

Keywords: Quantitative Structure-Activity Relationship. Binary Classification. Similarity Search. Virtual Screening. Machine Learning. Deep Machine Learning. Alzheimer's Disease.

LISTA DE ABREVIATURAS E SIGLAS

AChE	Acetilcolinesterase
AD	Domínio de aplicabilidade
API	*Application Programming Interface*, ou Interface de Programação de Aplicação
AUC	Área Sob a Curva
CID	Chemical Identifier, ou Identificador Químico
MCC	*Matthews Correlation Coefficient*, ou Coeficiente de Correlação de Matthews
MLP	*Multi-layer Perceptron*, ou Perceptron Multicamadas
OMS	Organização Mundial da Saúde
QSAR	*Quantitative Structure-Activity Relationship*, ou Relação Quantitativa entre Estrutura-Atividade
RF	*Random Forest*, ou Florestas Aleatórias
SVM	*Support Vectors Machine*, ou Máquina de Vetores Suporte
SiRMS	*Simplex Representation of Molecular Structure*, ou Representação Simples da Estrutura Molecular

SUMÁRIO

1 INTRODUÇÃO 17
1.1 Contextualização 17
1.2 Justificativa e motivação 18
1.3 Problema de pesquisa 18
1.4 Hipótese 19
1.5 Objetivos 19

2 FUNDAMENTAÇÃO TEÓRICA 21
2.1 Quimioinformática 21
2.1.1 Conceitos 21
2.1.2 Representação das estruturas 22
2.1.3 Bases de dados 24
2.1.4 Análise de similaridade química 26
2.1.5 Relações Quantitativas entre Estrutura Química e Atividade - QSAR 26
2.1.6 Descritores Moleculares 27
2.2 Aprendizado de Máquina 28
2.2.1 *Random Forest* 29
2.2.2 *Support Vector Machine* 30
2.2.3 *Multilayer Perceptron* 32
2.3 Aprendizado profundo de máquina: *TensorFlow* 33
2.3.1 Métodos de seleção de hiperparâmetros 35
2.3.1.1 Busca em Grade 36
2.3.1.2 Busca Aleatória 36
2.3.2 Validação de modelos QSAR 37
2.3.3 Avaliação de modelos QSAR 38
2.3.4 Domínio de aplicabilidade 41
2.4 Triagem Virtual 43

3 TRABALHOS RELACIONADOS 47
3.1 Estado da arte 47
3.2 Abordagens utilizadas para o desenvolvimento de modelos QSAR . 48
3.3 Outras aplicações 49

4 PROPOSTA DE SOLUÇÃO: TRIAGEM VIRTUAL UTILIZANDO CONSENSO ENTRE MODELOS QSAR E BUSCA POR SIMILARIDADE 51

4.1 Estruturação da metodologia empregada . . . 51
4.1.1 Etapa 01 - Preparação dos dados . . . 52
4.1.1.1 Definição do alvo químico / biológico / molecular . . . 52
4.1.1.2 Organização do conjunto de dados (conjunto de dados original) . . . 52
4.1.1.3 Avaliação da acurácia do conjunto de dados (conjunto de dados acurado) . 52
4.1.1.4 Cálculo dos descritores (variáveis/atributos) moleculares . . . 53
4.1.2 Etapa 02 - Construção dos modelos QSAR . . . 53
4.1.2.1 Conjuntos de dados . . . 53
4.1.2.2 Divisão do conjunto de dados em conjuntos de treinamento e teste . . . 54
4.1.2.3 Construção dos modelos usando os conjuntos de treinamento . . . 55
4.1.2.4 Validação dos modelos usando conjuntos de teste . . . 55
4.1.2.5 Seleção dos modelos para validação externa . . . 56
4.1.2.6 Teste de permutação . . . 57
4.1.3 Etapa 03 - Validação dos modelos . . . 57
4.1.3.1 Previsão de consenso da avaliação externa definida no Domínio de Aplicabilidade 57
4.1.4 Etapa 04 - Triagem virtual em bases de dados químicos . . . 59
4.1.4.1 Previsão de consenso dos compostos com os modelos obtidos . . . 59
4.1.4.2 Execução do procedimento de triagem virtual . . . 59

5 AVALIAÇÃO EXPERIMENTAL . . . 61
5.1 Preparação dos dados . . . 61
5.1.1 Definição do alvo químico . . . 61
5.1.2 Organização e avaliação da acurácia do conjunto de dados original . . . 62
5.1.2.1 Dados de treinamento e testes . . . 62
5.1.2.2 Dados para triagem virtual . . . 66
5.1.3 Seleção e cálculo dos descritores (variáveis) moleculares . . . 71
5.2 Construção dos modelos QSAR . . . 72
5.2.1 Modelagem do conjunto de dados . . . 72
5.2.2 Divisão do conjunto de dados em conjuntos de treinamento e teste . . . 74
5.2.3 Construção dos modelos usando os conjuntos de treinamento . . . 74
5.2.4 Validação dos modelos usando conjuntos de teste . . . 75
5.2.4.1 Descritores Morgan: Dados de treinamentos e testes . . . 75
5.2.4.2 Descritores SiRMS: Dados de treinamentos e teste . . . 81
5.2.4.3 Descritores *RDKit*: Dados de treinamentos e teste . . . 86
5.3 Seleção e validação dos modelos . . . 90
5.4 Triagem virtual em bases de dados químicos . . . 97
5.4.1 Execução do procedimento de triagem virtual . . . 97
5.4.2 Busca por similaridade . . . 100
5.4.3 Previsão de consenso dos compostos com os modelos obtidos . . . 102
5.5 Discussões . . . 105

5.5.1 Avaliação dos modelos em uma base de dados externa 107
5.5.2 Implicações práticas e potencial de aplicação 107

6 CONCLUSÕES 109

Referências 111

1 INTRODUÇÃO

1.1 Contextualização

Alzheimer é uma das doenças neurodegenerativas mais comuns e é uma das principais causas de demência em todo o mundo. Ela afeta, principalmente, a memória e outras funções cognitivas, como a capacidade de pensar, linguagem e tomar decisões. A sua prevalência tem aumentado em função do envelhecimento da população. A sua patologia é caracterizada pela formação de placas de beta-amiloide no cérebro, as quais podem interferir nas funções cognitivas e levar a perda progressiva de memória, dificuldades de comunicação, confusão, entre outros, e pode levar a uma completa dependência de cuidadores (DELANOGARE *et al.*, 2019).

Uma das abordagens terapêuticas para tratamento da doença de Alzheimer é a inibição da enzima acetilcolinesterase (AChE), que está diretamente relacionada à degradação da acetilcolina no cérebro, influenciando a função cognitiva. A acumulação de acetilcolina, resultante da inibição da AChE, tem o potencial de melhorar a transmissão neural, o que pode aliviar os sintomas associados à doença. Dessa forma, uma abordagem promissora para o tratamento dessa doença inclui a identificação de novos compostos que sejam capazes de inibir essa enzima (DHAMODHARAN; MOHAN, 2022).

Nesse contexto, os modelos QSAR (*Quantitative Structure-Activity Relationship*, ou Relação Quantitativa entre Estrutura-Atividade) têm se mostrado uma ferramenta importante para a descoberta de novos inibidores da AChE, fornecendo uma abordagem computacional eficiente e econômica para avaliar a atividade desses compostos (SHARMA; SHARMA, 2018).

Esses modelos conseguem prever a atividade biológica de compostos a partir de suas estruturas moleculares e, portanto, podem identificar novos candidatos a inibidores das enzimas de forma mais rápida e eficiente (PANTELEEV; GAO; JIA, 2018).

No entanto, ainda existem desafios a serem enfrentados para melhorar a precisão e a confiabilidade dos modelos QSAR. Um deles seria determinar quais descritores moleculares, algoritmos de aprendizado de máquina e aprendizado profundo de máquina são os mais adequados para criar modelos QSAR eficientes e precisos para a predição de atividade de inibidores da AChE (PATEL *et al.*, 2020; BAO *et al.*, 2023).

Adicionalmente, a criação de QSAR que combina diferentes métodos pode aumentar a precisão das previsões de atividade biológica de moléculas candidatas a fármacos. Essa abordagem pode incluir, por exemplo, o uso de diferentes tipos de descritores moleculares (como *Fingerprints* de Morgan, SiRMS e RDKit), diferentes métodos de seleção de descritores e diferentes algoritmos de aprendizado de máquina e aprendizado

profundo. A principal finalidade é aproveitar as vantagens de cada método para gerar um modelo de previsão de atividade biológica mais preciso e confiável, que possa ser utilizado no desenvolvimento de novos fármacos (JANG *et al.*, 2018).

1.2 Justificativa e motivação

Apesar de ser uma doença comum, não há uma cura definitiva para o Alzheimer e os tratamentos atuais podem apenas amenizar os sintomas, mas não impedem a progressão da doença. Segundo a Organização Mundial da Saúde (OMS), a doença de Alzheimer é a forma mais comum de demência, respondendo por cerca de 60 a 70% dos casos. Estima-se que cerca de 50 milhões de pessoas em todo o mundo tenham demência, e a cada ano são registrados cerca de 10 milhões de novos casos (ORGANIZATION, 2021).

Esses dados destacam a importância da busca por novas terapias para a doença de Alzheimer, e a identificação de novos compostos que possam inibir a AChE é uma das estratégias promissoras na luta contra essa doença (DHAMODHARAN; MOHAN, 2022).

A utilização de técnicas de aprendizado de máquina e aprendizado profundo têm se destacado como uma abordagem promissora na busca por novos compostos que possam auxiliar no tratamento da doença de Alzheimer. No entanto, a escolha adequada dos algoritmos e dos descritores moleculares utilizados é crucial para a obtenção de modelos precisos e confiáveis (NEVES *et al.*, 2018).

Outro fator fundamental é a validação dos modelos QSAR para garantir a eficácia e confiabilidade desses modelos na identificação de novos compostos com potencial atividade inibitória, e pode contribuir, significativamente, para o desenvolvimento de novas terapias para a doença de Alzheimer (CARPENTER; HUANG, 2018).

1.3 Problema de pesquisa

Esta pesquisa visa avaliar e combinar diferentes algoritmos de aprendizado de máquina e aprendizado profundo, utilizando diferentes tipos de descritores moleculares, na tarefa de prever a atividade de inibidor da AChE para a doença de Alzheimer. A escolha dos algoritmos e descritores moleculares adequados pode impactar na qualidade dos modelos gerados e na eficácia da identificação de novos compostos (DHAMODHARAN; MOHAN, 2022).

Compreender quais algoritmos e descritores moleculares são mais eficazes na tarefa de predição pode fornecer informações valiosas para os pesquisadores envolvidos. Outro ponto importante é que a escolha adequada de algoritmos e descritores moleculares pode reduzir os custos e o tempo necessários para identificar novos compostos, tornando o processo mais eficiente e econômico.

Assim, a pergunta de pesquisa que este estudo pretendeu responder foi: como diferentes abordagens de aprendizado de máquina, aprendizado profundo de máquina e descritores moleculares podem ser combinados para desenvolver modelos QSAR precisos e eficientes para a predição de atividade de inibidores da AChE para a doença de Alzheimer?

A resposta para essa pergunta tem potencial de contribuir com o desenvolvimento de novas terapias para o tratamento da doença de Alzheimer, a qual tem sido um grande desafio para a saúde pública.

1.4 Hipótese

Diante do exposto, este trabalho visou provar a seguinte hipótese: a criação de modelos QSAR, combinando diferentes abordagens de aprendizado de máquina, aprendizado profundo de máquina e descritores moleculares, resultará em uma maior capacidade de generalização e precisão na predição de atividade de inibidores da AChE para a doença de Alzheimer, em comparação com modelos QSAR criados com abordagens ou descritores isolados. Além disso, espera-se que a seleção dos melhores modelos, com base em seus hiperparâmetros dentro do domínio de aplicabilidade, possam ser utilizados como filtros moleculares durante triagem virtual, culminando na identificação de novos compostos promissores que possam aliviar os sintomas da doença de Alzheimer.

1.5 Objetivos

O objetivo geral deste estudo é realizar a triagem virtual para a identificação de novos compostos inibidores promissores, utilizando diferentes abordagens de aprendizado de máquina, aprendizado profundo e descritores moleculares como filtros moleculares.

Os objetivos específicos são:

- criar modelos QSAR usando diferentes abordagens (*Support Vector Machine*, *Random Forest*, *Multilayer Perceptron* e *TensorFlow Keras*) com diferentes descritores (*Fingerprints* de Morgan, SiRMS e RDKit) para a predição de atividade de inibidores da AChE para a doença de Alzheimer;
- avaliar a eficácia dos modelos QSAR desenvolvidos por meio da validação cruzada estratificada, usando métricas estatísticas apropriadas;
- realizar a busca por similaridade para identificação de compostos quimicamente semelhantes aos candidatos iniciais;
- selecionar os melhores modelos, com base em seus hiperparâmetros dentro do domínio de aplicabilidade, para serem utilizados como filtro molecular na etapa de triagem virtual.

- utilizar a abordagem de consenso dos resultados dos modelos e da busca por similaridade para identificação de novos compostos inibidores da AChE que possam ser promissores para o tratamento e prevenção da doença de Alzheimer durante a triagem virtual.

Para o alcance desses objetivos, este trabalho foi organizado nos seguintes capítulos:

- Capítulo 2.1.6: apresenta os fundamentos teóricos sobre a quimioinformática para compreender sobre modelos QSAR, as diferentes abordagens de aprendizado de máquina adotadas, análise de similaridade e triagem virtual.
- Capítulo 3: apresenta o estado arte sobre o tema de pesquisa.
- Capítulo 4: apresenta a proposta deste estudo, incluindo os métodos de como serão executados.
- Capítulo 5: apresenta os resultados encontrados.
- Capítulo 6: apresenta as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção é apresentado a fundamentação conceitual sobre as duas áreas envolvidas: quimioinformática e aprendizado de máquina.

2.1 Quimioinformática

2.1.1 Conceitos

Devido à ampliação do poder computacional das últimas décadas, assim como o crescimento exponencial da velocidade de geração de dados e a necessidade de lidar com grandes quantidades de informação química/biológica, o processo de descoberta de fármacos foi estimulado a englobar de maneira crescente em seus processos de pesquisa e desenvolvimento (P&D) abordagens tecnológicas. Isto culminou no fenômeno conhecido como explosão de dados ou *big data*. Neste contexto, o desenvolvimento de ferramentas capazes de extrair correlações e gerar modelos preditivos a partir de grandes volumes de informação tornou-se uma questão central neste processo (FERREIRA; ANDRICOPULO, 2018).

A quimioinformática é uma área interdisciplinar que utiliza recursos das ciências da computação e informação para resolver problemas da química (BUNIN *et al.*, 2007), os quais podem envolver diversos aspectos do processo de descoberta de candidatos a fármacos, assim como construção de modelos QSAR (modelos de aprendizado de máquina), mineração de dados (em bancos de dados químicos), mineração de grafos moleculares, dentre outros (SHARMA; SHARMA, 2018). A quimioinformática evoluiu muito nos últimos anos, desde técnicas de representação, manipulação e processamento de estruturas químicas até a análise e exploração de grandes bases de dados (LO *et al.*, 2018). Assim, a quimioinformática e a inteligência artificial têm estimulado o campo da descoberta e planejamento de candidatos a fármacos, sendo uma ferramenta indispensável para extrair informações químicas de grandes bases de dados de compostos, apoiando o desenvolvimento de fármacos de forma mais rápida e precisa (SHARMA; SHARMA, 2018; PANTELEEV; GAO; JIA, 2018).

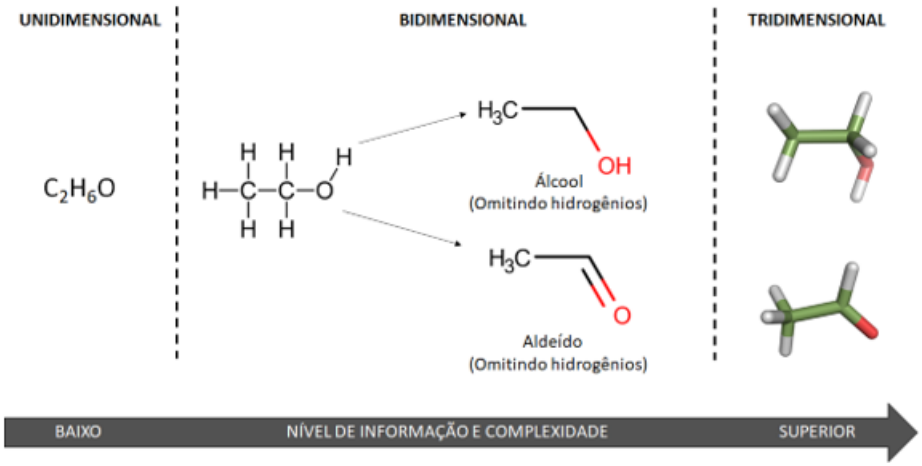
Vale destacar que a disponibilização dessas diversas e grandes bases de dados só se tornou possível com a produção e armazenamento de dados biológicos e químicos, produzidos pela química combinatória e por ensaios biológicos de alto desempenho (ZHU *et al.*, 2014). É importante frisar que analisar e explorar essas grandes bases de dados, de forma manual, se tornou inviável. Nesse contexto, a computação, com suas diversas ferramentas e técnicas, pode apoiar essa exploração, manipulação e processamento de estruturas químicas, gerando modelos computacionais capazes de fazer previsão e apoiar no

processo de descoberta e planejamento de novos candidatos a fármacos (CHEN; KOGEJ; ENGKVIST, 2018; LO *et al.*, 2018). Para tanto, estes compostos armazenados em bases de dados são representados utilizando estruturas computacionais, as quais apresentaremos na próxima seção.

2.1.2 Representação das estruturas

Um composto químico pode ser representado de diferentes formas gráficas para a compreensão humana. Isso se dá a partir da disposição e conexões de seus átomos podendo ser representados a partir das visualizações unidimensional (1D), bidimensional (2D) e tridimensional (3D), conforme ilustra a Figura 1.

Figura 1 – Diferentes níveis de representação molecular. Fonte: Autoria própria.



Entretanto, para a captura e processamento da informação referente às estruturas moleculares a partir de métodos computacionais, se faz necessária uma representação computacional de tradução da informação química para informação computacional por meio de noções lineares de representação de estruturas químicas. Portanto, para que o computador possa capturar, processar e compreender a estrutura química dos compostos, a mesma necessita estar descrita em uma sequência numérica única (ALVES *et al.*, 2018), caracterizada como uma assinatura digital exclusiva (INCHITRUST, 2020). As notações lineares mais conhecidas e utilizadas para codificar as estruturas químicas são (Figuras 2 e 3):

- SMILES (do inglês, *Simplified Molecular-Input Line-Entry System*)
- SMARTS (do inglês, *SMiles ARbitrary Target Specification*)
- InChIKey (do inglês, *International Union of Pure and Applied ChemistryKey*)

Figura 2 – Notação empregada para representar uma substância química no formato de InChI (*International Chemical Identifier*). Fonte: Autoria própria.

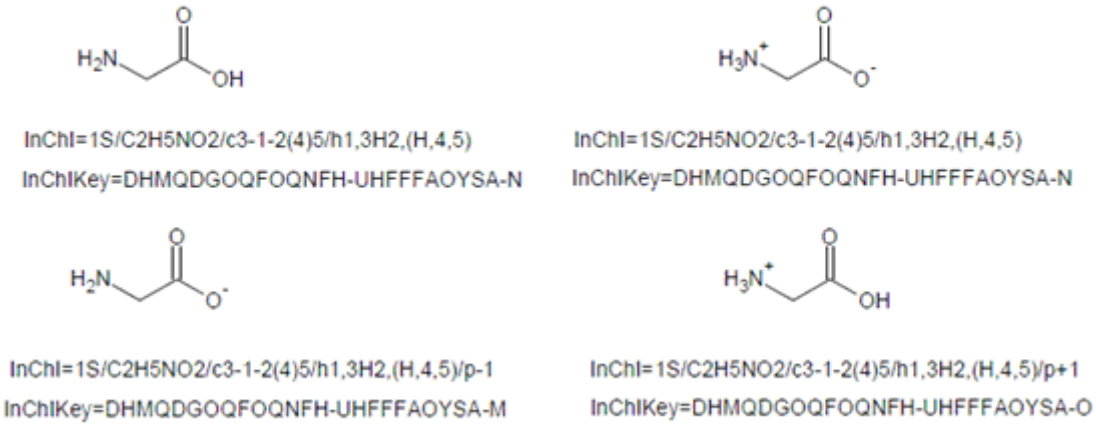
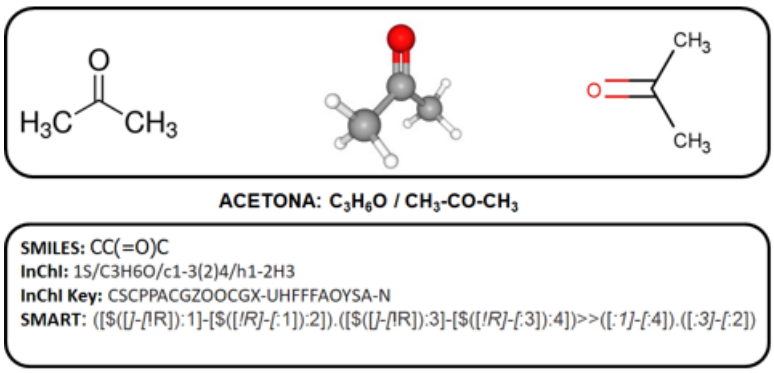
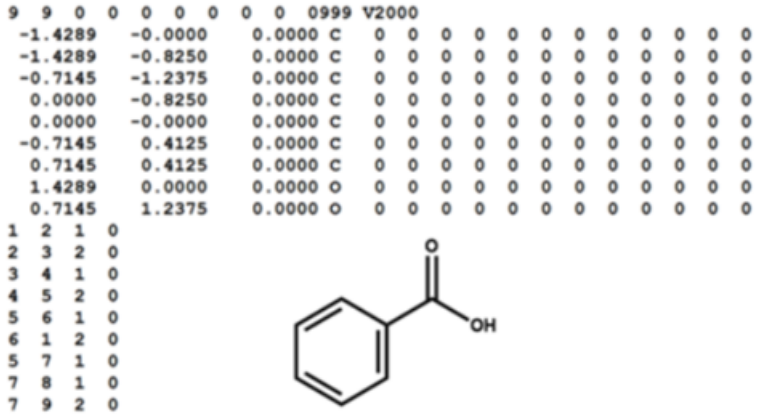


Figura 3 – Exemplo de notação SMILES, SMARTS, InChI e InChIKey. Fonte: Autoria própria.



Outros formatos também são utilizados para a representação molecular como os formatos CT (*Chemical ou Connection*), sendo MDL MOL (ou *molfile*) e MDL SDF (ou SD*file*) os mais utilizados. Esses formatos representam as estruturas químicas como se fossem grafos e as informações são armazenadas em uma tabela. A teoria dos grafos descreve a relação de objetos em determinado conjunto por meio de vértices. Em arquivos CT, átomos mais pesados que o hidrogênio correspondem aos vértices e ligações químicas às arestas (Figura 4).

Figura 4 – Representação molecular usando o formato MOL *file*. Fonte: Autoria própria.



Essas diferentes formas de representação das estruturas estão presentes em diversas bases de dados, as quais serão descritas no próximo tópico.

2.1.3 Bases de dados

Atualmente, existem várias bases de dados disponíveis em ambiente virtual que armazenam dados e informações relevantes para estudos de Química Medicinal. Elas fornecem informações químicas e biológicas de substâncias, como propriedades físico-químicas e resultados de ensaios *in vitro*, *in vivo* e, principalmente, resultados de triagem de alto desempenho (HTS)[1] (CHEN *et al.*, 2018). São exemplos de base de dados contendo informações químicas e biológicas de substâncias:

- **BMRDB** (*Biological Magnetic Resonance Data Bank*, www.bmrb.wisc.edu): é um banco de dados de espectroscopia de ressonância magnética nuclear em proteínas, peptídeos, ácidos nucléicos e outras biomoléculas (ULRICH *et al.*, 2008).

- **ChEMBL** (www.ebi.ac.uk/chembl): possui dados químicos, biológicos e genômicos, extraídos da literatura e de documentos de patentes, os quais podem ser usados para apoiar a tradução de informações genômicas em novos candidatos a fármacos. Ela possui 1.961.462 compostos e mais de 16.066.124 dados sobre atividades biológicas (EMBL-EBI, 2020).

- **DrugBank** (www.drugbank.ca): apresenta recursos que combinam dados detalhados sobre fármacos (produtos químicos, farmacológicos e farmacêuticos) e informações abrangentes sobre os alvos de fármacos (sequência e estrutura) (CHEN *et al.*, 2018). A versão mais recente, publicada em julho de 2020, contém 13.580 entradas de fármacos, incluindo 2.637 medicamentos aprovados (classificados como pequenas moléculas), 1.378 produtos biológicos aprovados (proteínas, peptídeos, vacinas e

[1] Do inglês *High Throughput Screening*.

alergênicos), 131 nutracêuticos e mais de 6.376 experimentais (em fase de descoberta) (WISHART *et al.*, 2018).

- **KEGG** (*Kyoto Encyclopedia of Genes and Genomes*, www.genome.jp/kegg/pathway .html): é uma base de dados que possui informações sobre genes e genomas para interpretação funcional, além de informações químicas e sistêmicas para aplicação prática de informações genômicas (KANEHISA *et al.*, 2019).

- **PDB** (*Protein Data Bank*, www.rcsb.org): apresenta dados sobre biologia molecular, estrutural e computacional, incluindo dados sobre as formas 3D de proteínas e ácidos nucléicos (BERMAN; HENRICK; NAKAMURA, 2003).

- **PubChem** (pubchem.ncbi.nlm.nih.gov): é uma das maiores bases de dados que fornece informações sobre substâncias químicas e suas atividades biológicas, envolvendo algumas subcategorias, como substância, composto e *BioAssay*. Além disso, é possível encontrar informações sobre segurança e toxicidade, patentes, referências dentre outras (NIH, 2020).

- **STITCH** (*Search Tool for Interactions of Chemicals*, stitch.embl.de): é uma base de dados que fornece uma rede de interações química-proteína conhecidas e previstas. As interações incluem associações diretas (físicas) e indiretas (funcionais); decorrem de previsão computacional, de transferência de conhecimento entre organismos e de interações agregadas de outros bancos de dados (primários). São mais de 2031 organismos identificados, 9,6 milhões de dados sobre proteínas e 1,6 bilhões de informações sobre interações (SZKLARCZYK *et al.*, 2016).

- **SuperPred** (prediction.charite.de): é uma base de dados que possui informações sobre interações entre composto-alvo, conectando similaridade química de compostos semelhantes a fármacos com alvos moleculares e abordagem terapêutica semelhantes (NICKEL *et al.*, 2014).

Além destas, existem outras bases de dados que podem ser utilizadas em estudos de química medicinal, tais como ASDCD (*Antifungal Synergistic Drug Combination Database*), BRENDA (*The Comprehensive Enzyme Information System*), CancerDR (*Cancer Drug Resistance Database*), DCDB (*Drug Combination Database*), MATADOR (*Manually Annotated Targets and Drugs Online Resource*), BindingDB (*The Binding Database*), SuperTarget, TDR *targets* e *Therapeutic Target Database* (CHEN *et al.*, 2018).

Essas bases de dados, em muitos casos, são utilizadas para a identificação de compostos similares. No próximo tópico abordaremos a relevância da análise de similaridade química e suas aplicações nos estudos de química medicinal.

2.1.4 Análise de similaridade química

A semelhança das propriedades entre as moléculas ou similaridade química é um dos conceitos mais explorados na quimioinformática. A similaridade química é importante para estabelecer relações entre estrutura e atividade ou propriedade (QSAR ou QSPR) e compreender o comportamento de determinado grupo de moléculas (MAGGIORA *et al.*, 2014).

A similaridade química contribui para encontrar erros experimentais ou *cliffs*, pares de estruturas químicas semelhantes com atividade/propriedade muito diferentes em um subgrupo de moléculas (GUHA; DRIE, 2008). Empregando métodos computacionais, a similaridade é calculada aplicando-se uma função de similaridade (também chamada de coeficiente de similaridade) com base nos descritores moleculares. Dentre as funções de similaridade mais utilizadas, podem ser citados o coeficiente de Tanimoto, e as distâncias Euclidiana e de Mahalanobis (MAGGIORA *et al.*, 2014).

Qualquer tipo de descritor pode ser utilizado na análise de similaridade, mas os descritores baseados em fragmentos moleculares, principalmente os do tipo impressão digital ou *fingerprints*, são os mais utilizados por serem mais fáceis de interpretação (WILLETT, 2006).

Como dito anteriormente, a similaridade química é importante para se estabelecer relações entre estrutura e atividade ou propriedade (QSAR ou QSPR) e também compreender o comportamento de determinado grupo de moléculas (MAGGIORA *et al.*, 2014), como será abordado no próximo tópico.

2.1.5 Relações Quantitativas entre Estrutura Química e Atividade - QSAR

A relação entre a estrutura química e a propriedade biológica ou propriedade físico-química pode ser modelada por uma equação matemática, que pode ser chamada de relação quantitativa estrutura-atividade (QSAR). Esta área tem como principal abordagem a aplicação de diversos métodos estatísticos de análise de dados com o intuito de desenvolver modelos que possam predizer corretamente determinada propriedade biológica de compostos baseados em sua estrutura química. Para se estabelecer essa relação, é necessário o cálculo de descritores moleculares e dados biológicos definidos experimentalmente (TROPSHA *et al.*, 2017). Como resultado, o modelo QSAR pode ser representado por meio da seguinte equação:

$$P_i = k^{'}(D_1, D_2 \cdots, D_n) \tag{2.1}$$

em que, P_i é uma variável dependente que representa valores previstos da resposta biológica; $k^{'}$ são coeficientes de ajustes aplicados nas variáveis independentes; e, $D_1, D_2 \cdots, D_n$ são variáveis independentes, também chamadas de variáveis descritivas, e indicam as propriedades referentes a valores que representam cada descritor molecular.

Estudos de QSAR apresentam várias aplicações na área de planejamento de candidatos a fármacos, tais como (i) identificação de novos ligantes/protótipos com atividade/propriedade desejada; (ii) otimização da atividade/propriedade; e (iii) identificação de compostos com efeitos potencialmente indesejados em estágios preliminares do desenvolvimento (TROPSHA, 2010).

Com efeito, o crescente desenvolvimento das ciências "ômicas", aliado ao aprimoramento de recursos computacionais, ao progressivo aumento da disponibilidade de conjuntos de dados de alta qualidade e ao desenvolvimento de modelos preditivos, pode-se dizer que o campo de estudos de QSAR e suas aplicações ainda é um campo fértil para pesquisas na área de química medicinal (FOURCHES, 2014; GORB; KUZ'MIN; MURATOV, 2014).

Como apresentado anteriormente, os descritores moleculares são partes fundamentais na geração de modelos de aprendizado de máquina. Abordaremos, a seguir, diferentes estratégias computacionais para a obtenção de descritores moleculares.

2.1.6 Descritores Moleculares

Um descritor molecular é o resultado final de um procedimento matemático e lógico que transforma informação química codificada em uma representação simbólica de uma molécula em um número útil ou o resultado de algum experimento padronizado. Descritores moleculares contribuem para a compreensão de propriedades moleculares e/ou podem ser utilizados na geração de um modelo matemático para a previsão de determinada propriedade de outras moléculas (TODESCHINI; CONSONNI, 2000).

Diferentes tipos de descritores químicos refletem diferentes níveis de representação estrutural. Esses descritores podem ser classificados quanto à sua "dimensionalidade" em unidimensionais (1D), baseados em propriedades físico-químicas e fórmula molecular (por exemplo, massa molecular, refratividade molar, logP, entre outros); bidimensionais (2D), que descrevem propriedades que podem ser calculadas a partir de uma representação 2D (tais como número de átomos, número de ligações, índices de conectividade, entre outros); e tridimensionais (3D), que dependem da conformação das moléculas (por exemplo, volume de van der Waals, área de superfície acessível ao solvente, entre outros) (XUE; BAJORATH, 2000). Para desenvolver modelos de QSAR/QSPR, descritores e dados de atividade/ propriedade são armazenados em uma tabela (Tabela 1). Nela, os dados de atividade/propriedade são armazenados na matriz Y e os descritores na matriz X. Vários tipos de relações podem ser obtidos a partir dessas matrizes. Modelos QSAR podem ser gerados de forma categórica (por exemplo, ativos/inativos, tóxico/não tóxico) ou uma relação quantitativa, na qual a propriedade estudada (Y) é representada por uma função de um ou mais descritores (X) (PUZYN; LESZCZYNSKI; CRONIN, 2010).

A construção dos modelos, como apresentado anteriormente, vem da regressão que relaciona um conjunto de atributos (descritores) de um composto químico e a sua

Tabela 1 – Relação entre dados de atividade/propriedade e descritores moleculares.

Identificador químico	**Atividade / propriedade**	**Descritor 1**	**Desc. 2**	**Desc. 3**	**...**	**Descr. n**
Molécula 1	Y_1	X_{11}	X_{12}	X_{13}	...	X_{1n}
Molécula 2	Y_2	X_{21}	X_{22}	X_{23}	...	X_{2n}
Molécula 3	Y_3	X_{31}	X_{32}	X_{33}	...	X_{3n}
Molécula 4	...	...	...	...	...	...
Molécula 5	Y_n	X_{n1}	X_{n2}	X_{n3}	...	X_{nn}

atividade biológica com relação a um ou mais alvos biológicos. Para realizar essa tarefa, algoritmos de aprendizado de máquina podem ser utilizados, os quais serão apresentados na próxima seção.

Nesta seção serão apresentados os fundamentos relacionados às diferentes abordagens de aprendizado de máquina, aprendizado profundo de máquina, QSAR, descritores moleculares, análise de similaridade e triagem virtual.

2.2 Aprendizado de Máquina

Aprendizado de máquina é uma técnica bastante utilizada para processar grandes quantidades de dados e extrair visões (*insights*) valiosos. Os quatro principais paradigmas de algoritmos de aprendizado de máquina encontrados na literatura são: supervisionado, não-supervisionado, semi-supervisionado e por reforço (Figura 5) (MITCHELL, 1997).

Figura 5 – Tipos de aprendizado de máquina. Fonte: Autoria própria.

No aprendizado supervisionado, o algoritmo recebe dados de entrada e saída rotulados, permitindo que seja estabelecido um "mapeamento" entre eles. Já no aprendizado não-supervisionado, os dados de entrada não têm rótulos, e o algoritmo deve identificar padrões ou estruturas por meio desses dados (MITCHELL, 1997).

No aprendizado semi-supervisionado, a maioria dos dados de entrada não tem rótulos, mas pode haver alguns dados rotulados também. Assim, os dados rotulados são usados para obter mais informações sobre os dados e, dessa forma, realizar o processo de

aprendizado tendo como base os dados não rotulados. Isso permite que o algoritmo use dados rotulados e não rotulados para realizar tarefas supervisionadas ou não supervisionadas (BRUCE, 2001).

No aprendizado por reforço, os modelos são treinados para tomar decisões em um ambiente incerto e complexo, recebendo recompensas ou penalidades com base nas ações tomadas. A tentativa e erro é usada para maximizar as recompensas (GOODFELLOW; BENGIO; COURVILLE, 2016).

Adicionalmente, o desenvolvimento e a implementação de métodos de aprendizado de máquina podem auxiliar, consideravelmente, o processo de descoberta precoce de candidatos a fármacos, especialmente para a doença de Alzheimer. Nas subseções seguintes, abordaremos os algoritmos de aprendizado de máquina adotados neste trabalho: *Random Forest*, SVM, *Multilayer Perceptron* e a biblioteca *TensorFlow*.

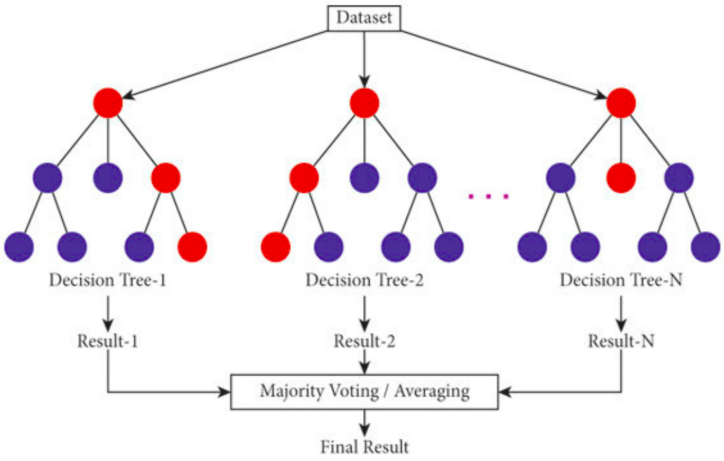
2.2.1 *Random Forest*

A *Random Forest*, ou Florestas Aleatórias em português, é um método de aprendizado supervisionado que pode ser usado para solucionar problemas de classificação e regressão. É uma combinação de várias árvores de decisão, em que cada árvore é construída a partir de uma amostra aleatória (com reposição) do conjunto de dados original (SVETNIK *et al.*, 2003).

Florestas aleatórias são usadas para prever um valor ou propriedade de interesse: regressão (contínuo) ou classificação (categórico). Em um problema de classificação, o algoritmo gera várias árvores de decisão a partir do conjunto de dados de treinamento e a saída é definida pela votação majoritária. Já em problemas de regressão, o valor final é calculado como a média das previsões de cada árvore para cada observação (HORVATH; ALDAHDOOH, 2017).

Dessa forma, nesse método, cada árvore é construída de forma independente, a partir de uma amostra *bootstrap* dos dados originais. Assim, dois terços dos exemplos originais são utilizados na construção de cada árvore (k-ésima). O conjunto restante é usado para avaliação do erro (BREIMAN, 2001). Isso ajuda a reduzir a correlação entre as árvores e melhora a capacidade de generalização do modelo.

Figura 6 – Ilustração das árvores na florestas aleatórias. Fonte: www.researchgate.net.



O processo de construção das árvores inclui uma divisão aleatória dos dados em um conjunto de treinamento e um conjunto de testes; a construção da árvore de classificação a partir do conjunto de treinamento; e a comparação entre a classe prevista e a classe verdadeira para cada elemento do conjunto de testes. Como exemplificado na figura 6, O procedimento é repetido várias vezes para gerar as árvores de classificação. O resultado final gerado contempla as taxas médias de erro sobre as árvores, assim como os respectivos erros-padrão (BREIMAN, 2001).

Cada árvore é construída a partir de uma amostra aleatória dos dados originais e uma seleção aleatória de um subconjunto de variáveis. A melhor divisão é usada para dividir cada nó. As árvores são crescidas ao máximo, sem poda.

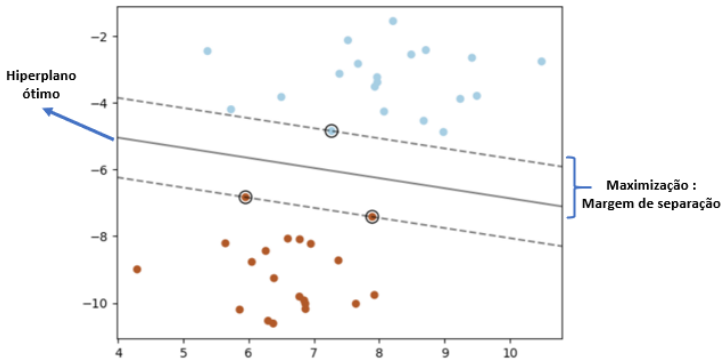
Destaca-se, que a taxa de erro de uma floresta de árvores de decisão depende da robustez das árvores individuais na floresta e da correlação entre suas classificações (BREIMAN, 2001).

2.2.2 *Support Vector Machine*

Support Vector Machine (SVM), ou Máquinas de Vetores Suporte em português, são um algoritmo de aprendizado de máquina supervisionado utilizado para resolver problemas de classificação e regressão. Esse algoritmo é capaz de realizar tanto classificação linear quanto uma classificação não linear, graças a uma eficiente abordagem conhecida como truque do *kernel* (JAMES *et al.*, 2017).

Para o caso de classificação binária, o SVM busca encontrar o hiperplano de separação ótimo que maximize a margem entre duas classes (JAMES *et al.*, 2017). Em outras palavras, o SVM procura a linha que melhor separa os dados em dois grupos, onde cada grupo representa uma classe diferente (Figura 7) (GULIA; DHAIYA; ANSHUL, 2019). Esse hiperplano é construído utilizando técnicas de programação quadrática, e foi proposto originalmente por Boser e Vapnik (SUSHKO, 2011).

Figura 7 – Hiperplano ótimo separando os dados com a máxima margem. Os vetores-suporte estão circulados em preto. Fonte: Baseado em (GULIA; DHAIYA; ANSHUL, 2019).



Ressalta-se que o SVM não é limitado apenas a problemas de classificação binária, podendo ser estendido para classificar em múltiplas classes. Além disso, ele pode ser usado para problemas de regressão, em que o objetivo é encontrar a melhor linha ou superfície para se ajustar aos dados. Dessa forma, em problemas de classificação com mais de duas classes, o SVM pode ser aplicado de duas maneiras principais: um-contra-um (*one-vs-one*) ou um-contra-todos (*one-vs-all*) (YANG *et al.*, 2013).

No método um-contra-um, o SVM cria um modelo para cada par de classes e faz a classificação a partir da votação da maioria dos modelos. Por exemplo, se houver 4 classes (A, B, C e D), serão criados seis modelos: A vs. B, A vs. C, A vs. D, B vs. C, B vs. D e C vs. D. Cada modelo irá gerar uma decisão de classificação e a classe mais votada será a classe final do objeto (GONCALVES, 2008).

Já no método um-contra-todos, o SVM treina um modelo para cada classe em relação a todas as outras. Por exemplo, se houver 4 classes (A, B, C e D), serão criados quatro modelos: A vs. BCD, B vs. ACD, C vs. ABD e D vs. ABC. Cada modelo irá gerar uma decisão de classificação e a classe com o maior valor de confiança será escolhida como a classe final do objeto (FRIEDMAN, 1996; GONCALVES, 2008).

Destaca-se que ambos os métodos têm suas vantagens e desvantagens. No método um-contra-um, há menos dados de treinamento para cada modelo, resultando em modelos mais precisos e rápidos. Por outro lado, a construção de um grande número de modelos pode ter alto custo computacional. No método um-contra-todos, há mais dados de treinamento para cada modelo e isso pode resultar em modelos mais robustos. Porém, o desempenho pode ser afetado quando as classes são desbalanceadas (FRIEDMAN, 1996).

Os pontos fortes do SVM incluem a capacidade de lidar com conjuntos de dados de alta dimensionalidade e a habilidade de generalizar bem para novos dados, isto é, ele pode fazer previsões precisas em conjuntos de dados que nunca viu antes. Entretanto, o SVM pode ser sensível à escolha de parâmetros, como o tipo de kernel escolhido pode afetar

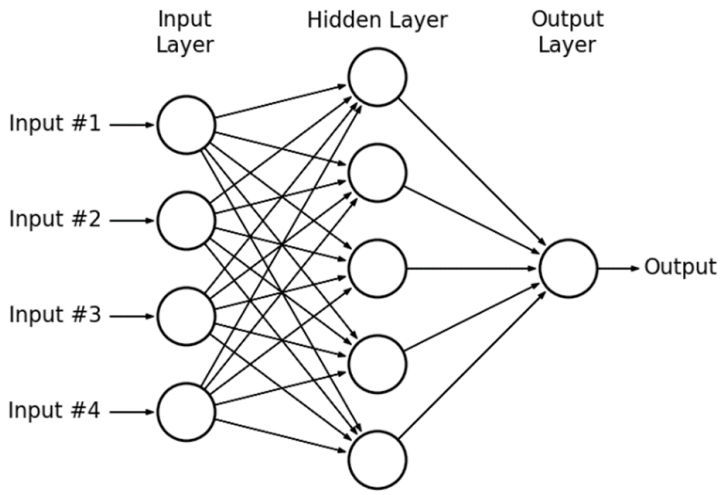
significativamente o desempenho do algoritmo. Além disso, a largura de banda (conhecida como parâmetro de regularização) também pode influenciar a qualidade das previsões. Por isso, é necessário ajustar esses parâmetros para garantir que o SVM esteja funcionando da melhor forma possível para o conjunto de dados em questão (GULIA; DHAIYA; ANSHUL, 2019).

2.2.3 *Multilayer Perceptron*

O *Multilayer Perceptron* (MLP) é uma rede neural artificial composta por camadas de neurônios que processam informações de entrada e geram saídas por meio de um processo de aprendizado supervisionado. Esse algoritmo tem sido aplicado com sucesso em uma série de problemas difíceis, principalmente para problemas de classificação e regressão (HAYKIN, 2009).

Desta forma, um MLP consiste em um conjunto de unidades sensoriais (nós fontes) que constituem a camada de entrada; uma ou mais camadas ocultas (ou intermediárias) de nós computacionais (neurônios); e uma camada de saída de nós computacionais (neurônios) estas camadas podem ser exemplificadas na figura 8. Adicionalmente, se faz necessário esclarecer alguns conceitos, dentre eles (GARDNER; DORLING, 1998):

Figura 8 – Camadas do Multilayer Perceptron. Fonte: https://www.nomidl.com/natural-language-processing/what-is-multilayer-perceptron/



- neurônio: é a unidade básica de processamento do MLP. Ele recebe entradas ponderadas, as soma e aplica uma função de ativação para produzir uma saída.

- camada: é um conjunto de neurônios que processam informações de entrada, de forma paralela. Existem três tipos de camadas no MLP: a camada de entrada, a camada oculta e a camada de saída.

- pesos: são valores atribuídos a cada entrada do neurônio, determinando a influência que cada entrada terá na saída final. Os pesos são ajustados durante o processo de treinamento da rede para minimizar o erro de predição.
- função de ativação: é aplicada à soma ponderada das entradas do neurônio para determinar a saída do neurônio. Ela é responsável por introduzir não linearidade na rede e permitir que ela aprenda relações complexas entre as entradas e as saídas.
- *feedforward*: é o processo de propagar as entradas da rede através das camadas até a camada de saída, produzindo uma saída final.
- *backpropagation*: é o algoritmo de treinamento do MLP, que consiste em propagar o erro de predição da saída da rede de volta através das camadas até a camada de entrada, ajustando os pesos ao longo do caminho para minimizar o erro.
- *overfitting*: é um problema comum no treinamento do MLP, que ocorre quando a rede se ajusta demais aos dados de treinamento e não generaliza bem para novos dados. Isso pode ser evitado através de técnicas como a regularização e a validação cruzada.

Portanto, o algoritmo geralmente mais utilizado para treinar uma rede MLP é o de retropropagação (*Backpropagation*). Esse algoritmo foi desenvolvido por Rumelhart, Hinton e Willians, em 1986, e é composto por quatro passos: inicialização, ativação, treinar pesos e iteração. Esse algoritmo ajusta os pesos da rede para minimizar o erro entre a saída real e a saída prevista, sendo que cada entrada de treinamento está associada a uma saída desejada. Assim, o MLP pode relacionar o conhecimento a vários neurônios de saída (HAYKIN, 2001).

Os pontos fortes do MLP incluem a sua flexibilidade (capacidade de lidar com vários problemas, desde as tarefas de classificação binária até problemas de regressão e classificação multiclasse), sua capacidade de lidar com problemas não-lineares, escalabilidade (capacidade de lidar com grandes conjuntos de dados e alta dimensionalidade) e sua habilidade de generalização (GERTRUDES *et al.*, 2012). No entanto, o MLP pode ser sensível à escolha do número de camadas ocultas, do número de neurônios em cada camada e da taxa de aprendizado. A escolha adequada desses parâmetros é essencial para obter um bom desempenho do MLP (QUADRI *et al.*, 2022).

2.3 Aprendizado profundo de máquina: *TensorFlow*

O aprendizado profundo de máquina, conhecido como *deep learning*, é uma subárea da Inteligência Artificial que se destaca por sua capacidade de aprender automaticamente recursos complexos e robustos a partir de dados brutos, sem a necessidade de engenharia de recursos. Atualmente, existem diversas bibliotecas utilizadas para o desenvolvimento

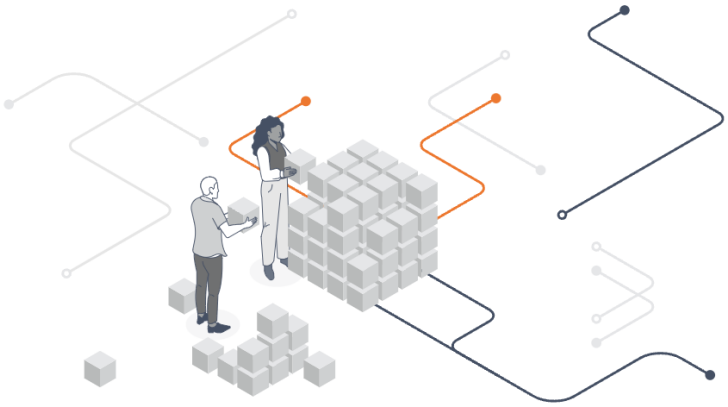
de modelos de aprendizado profundo, como PyTorch, Theano, Keras, *TensorFlow*, dentre outras (IQBAL *et al.*, 2023).

TensorFlow é uma biblioteca de código aberto, desenvolvida pela equipe do *Google Brain* (GOOGLE, 2023), que se tornou uma das mais populares, de acordo com as avaliações com estrelas no GitHub. Ela é considerada a mais fácil de usar, fornecendo uma estrutura flexível e robusta para criar, treinar e implantar modelos de aprendizado profundo em uma variedade de domínios, desde a visão computacional até o processamento de linguagem natural, a descoberta de novos medicamentos, etc (ALZUBAIDI *et al.*, 2021).

TensorFlow é conhecido por seu modelo de programação orientado a grafos, sendo possível representar as operações matemáticas como nós em um grafo direcionado, em que os dados fluem através das arestas desse grafo, por isso a origem do nome *TensorFlow*. Essa abordagem permite que os desenvolvedores definam a arquitetura de um modelo de maneira abstrata e, em seguida, otimizem de forma eficiente a execução do modelo em hardware. Além disso, a estrutura de grafos proporciona que o *TensorFlow* seja altamente escalável e adequado para o treinamento distribuído em *clusters* de computadores (TENSORFLOW, 2023a).

Os modelos de aprendizado profundo, no *TensorFlow*, são construídos usando camadas, as quais são blocos fundamentais que podem ser empilhados para criar arquiteturas complexas de redes neurais (Figura 9). Ele fornece várias camadas: densas (*fully connected*), convolucionais, recorrentes, etc, as quais facilitam a construção de arquiteturas personalizadas para as tarefas específicas, como classificação de imagens, tradução de idiomas e detecção de objetos (ABADI *et al.*, 2016).

Figura 9 – Ilustração do *TensorFlow*, disponível em www.tensorflow.org/.



Dentre as várias classes da biblioteca *TensorFlow*, destaca-se a "*tf.keras.Sequential*" que permite criar modelos de redes neurais sequenciais de forma mais simples. Nela, as camadas são empilhadas uma após a outra na ordem em que são adicionadas ao modelo, tornando-a ideal para modelos lineares e simplificando a construção de arquiteturas de

aprendizado profundo para tarefas como classificação, regressão e demais desafios de aprendizado de máquina. Essa classe faz parte do módulo "*tf.keras*", que é uma API (*Application Programming Interface*) de alto nível para construir e treinar modelos de aprendizado profundo em *TensorFlow* (TENSORFLOW, 2023b).

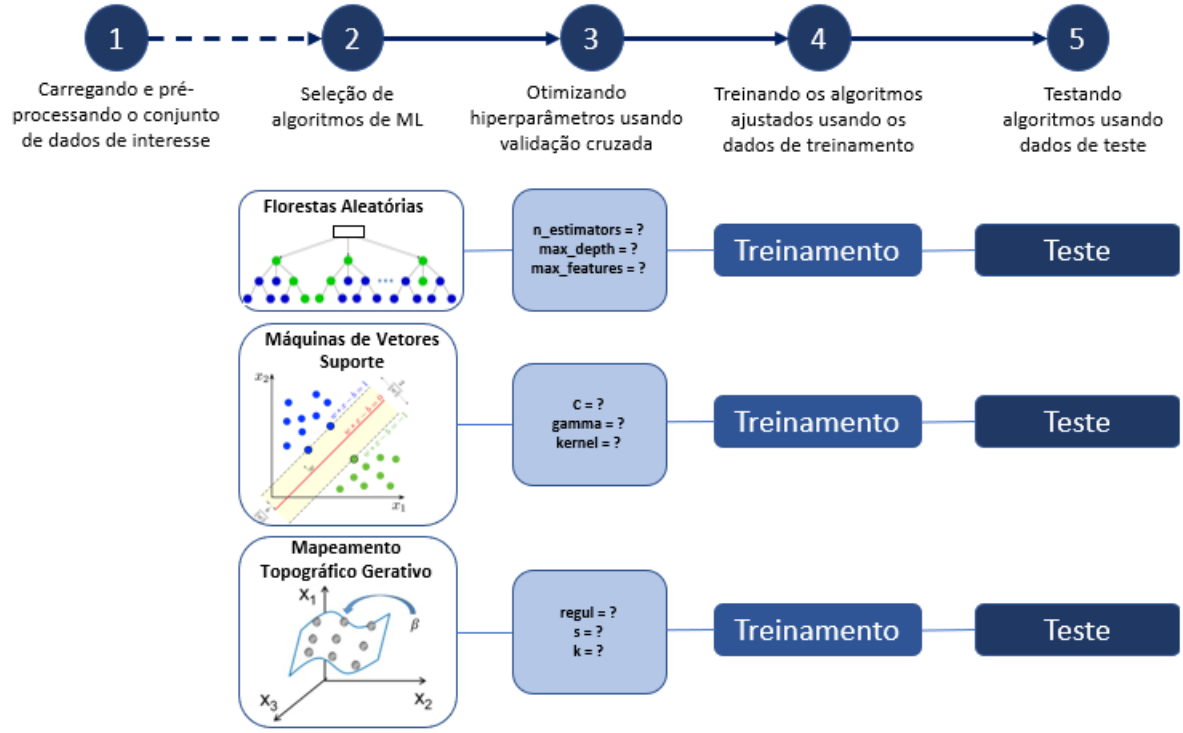
2.3.1 Métodos de seleção de hiperparâmetros

A maioria dos algoritmos de aprendizado de máquina contém, pelo menos, um hiperparâmetro para controlar a complexidade do modelo. A escolha dos valores para os hiperparâmetros influencia o desempenho do modelo, sendo ainda considerado um desafio computacional (PEDREGOSA, 2016). A seleção dos hiperparâmetros é importante por apresentar os seguintes efeitos (FEURER; HUTTER, 2019):

- reduzir o esforço humano necessário para aplicar métodos de aprendizado de máquina;
- melhorar o desempenho de algoritmos de aprendizado de máquina, adaptando-os ao problema em questão.
- melhorar a reprodutibilidade dos estudos científicos. Isso facilita comparações justas, uma vez que métodos diferentes só podem ser comparados de forma justa se todos eles receberem o mesmo nível de ajuste para o problema em questão.

A Figura 10 ilustra as fases de otimização para alguns exemplos de hiperparâmetros, de acordo com o tipo de algoritmo.

Figura 10 – Exemplos de cenários para seleção de hiperparâmetros. Fonte: Autoria própria.



Logo, existem dois tipos de métodos de otimização de hiperparâmetros: pesquisa manual e métodos de pesquisa automatizada. O primeiro, por testar manualmente os conjuntos de hiperparâmetros, depende da intuição e experiência de usuários especialistas que podem identificar os parâmetros importantes que têm um maior impacto nos resultados e, assim, determinar a relação entre determinados parâmetros e os resultados finais por meio das ferramentas de visualização. Isso requer que os usuários tenham mais conhecimento profissional e experiência prática sendo, portanto, difícil de ser aplicado por usuários não especialistas (WU *et al.*, 2019). O segundo método visa superar as desvantagens da busca manual, propondo algoritmos de busca automatizada, tais como busca em grade e busca aleatória. Os algoritmos de busca automatizada serão detalhados nos tópicos a seguir.

2.3.1.1 Busca em Grade

A busca em grade[2] é uma abordagem de busca de parâmetros que gera, exaustivamente, candidatos a partir de uma grade de valores ideais. Essa abordagem analisa cada combinação de valores possíveis de hiperparâmetros. Em seguida, avalia o desempenho de acordo com uma métrica pré-definida pelo método de validação cruzada. Por último, são obtidos valores de hiperparâmetros que alcançam o melhor desempenho (LIASHCHYNSKYI; LIASHCHYNSKYI, 2019).

A busca em grade é basicamente uma lista de valores candidatos para cada hiperparâmetro. O nome "grade" vem do fato de que todos os candidatos possíveis dentro de todos os hiperparâmetros necessários são combinados em uma espécie de grade. A combinação que produz o melhor desempenho, preferencialmente avaliada em um conjunto de validação é, então, selecionada (BERGSTRA; BENGIO, 2012).

Vale ressaltar que a eficiência deste algoritmo diminui conforme aumenta o número de hiperparâmetros ou o aumento da faixa de valores a serem ajustados (BERGSTRA; BENGIO, 2012). Portanto, o uso da pesquisa em grade é indicado a depender do número de possibilidades de ajustes (LIASHCHYNSKYI; LIASHCHYNSKYI, 2019).

2.3.1.2 Busca Aleatória

O algoritmo de busca aleatória[3] tenta combinações aleatórias a partir de uma gama de valores, em que cada configuração é amostrada a partir de uma distribuição de possíveis valores de parâmetro. Em comparação com o algoritmo de pesquisa em grade, a pesquisa aleatória é mais eficiente em um espaço de alta dimensão, embora não seja confiável para treinar modelos complexos (WU *et al.*, 2019).

A seleção dos valores a serem avaliados é totalmente aleatória. Além da velocidade, a pesquisa aleatória aproveita-se da aleatoriedade no caso de hiperparâmetros contínuos

[2] Do inglês *Grid Search*.
[3] Do inglês *Random Search*.

que devem ser discretizados quando otimizados pela pesquisa em grade (BERGSTRA; BENGIO, 2012).

Após a apresentação de algumas características importantes relacionadas com os algoritmos empregados neste trabalho, a seguir, será detalhada a etapa de validação de modelos de aprendizado de máquina.

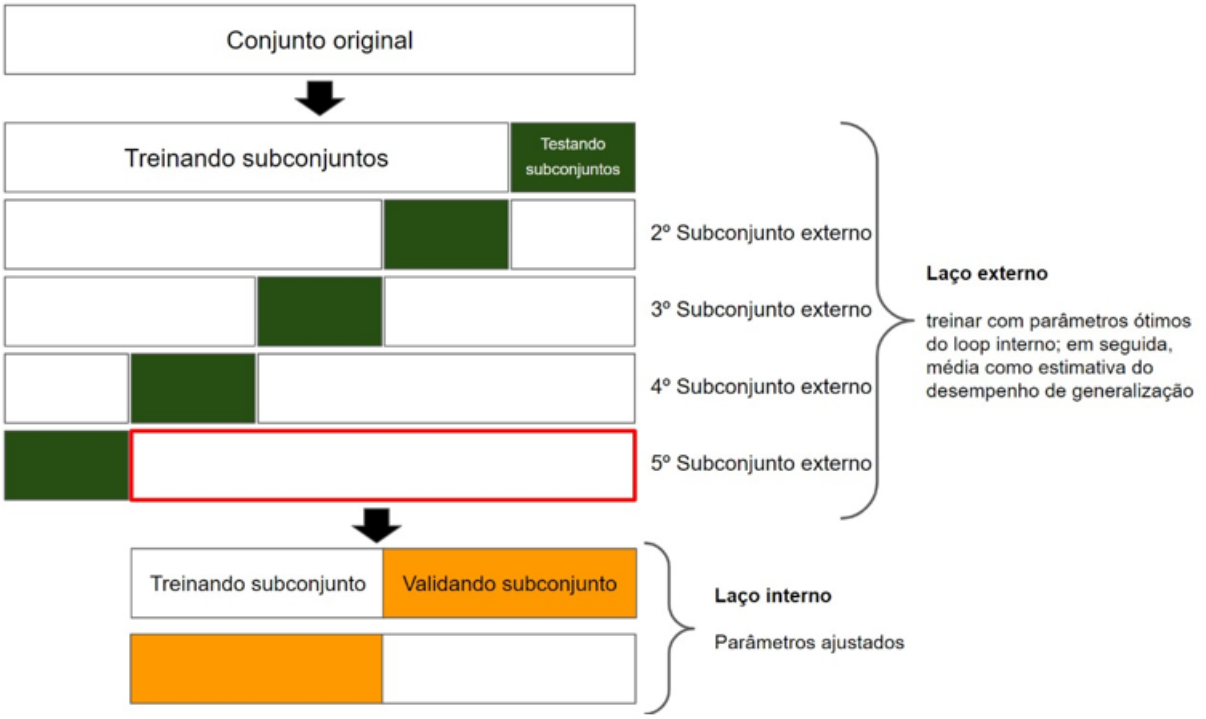
2.3.2 Validação de modelos QSAR

Percebe-se na literatura que métodos para validação de modelos QSAR propõem a divisão do conjunto de amostras em dois subconjuntos, com tamanhos diferentes: geralmente, 70% e 30% (MAZZOLARI; VISTOLI, 2015), ou 80% e 20% (TROPSHA, 2010; TROPSHA *et al.*, 2017). O conjunto maior é responsável por treinar ("conjunto de treinamento") e o menor por testar ("conjunto de teste") o modelo. Essa subdivisão pode ser realizada aleatoriamente, para cada um dos subconjuntos. Dessa forma, a depender da quantidade de alvos e compostos, todos estarão presentes nos conjuntos de treinamento e teste (JUNG, 2018).

A validação do modelo acontece por meio da "Validação Cruzada Aninhada" (NCV)[4], conforme ilustrado na Figura 11. Esse método executa dois laços (*loops*) aninhados, sendo o primeiro responsável pela validação externa e executado logo após o *loop* interno ser concluído. Esse *loop* interno fornece os melhores parâmetros e é conhecido como validação interna (PARVANDEH *et al.*, 2020).

[4] Do inglês *Nested Cross Validation*.

Figura 11 – Representação do método de validação cruzada aninhada, sendo $k = 5$. Fonte: Autoria própria.



Dentre os diferentes tipos de validação interna, um método bastante empregado é o da validação cruzada *k-fold* (TROPSHA, 2010; CHERKASOV *et al.*, 2014). Ele subdivide o conjunto de treinamento em k subconjuntos de tamanhos iguais. Em seguida, treina o modelo em subconjuntos k-1 e, depois, testa o modelo no subconjunto restante. Esse processo é repetido k vezes, alterando os elementos do conjunto de teste, possibilitando que todos os k subconjuntos tenham feito parte do conjunto de teste (OJALA; GARRIGA, 2010).

2.3.3 Avaliação de modelos QSAR

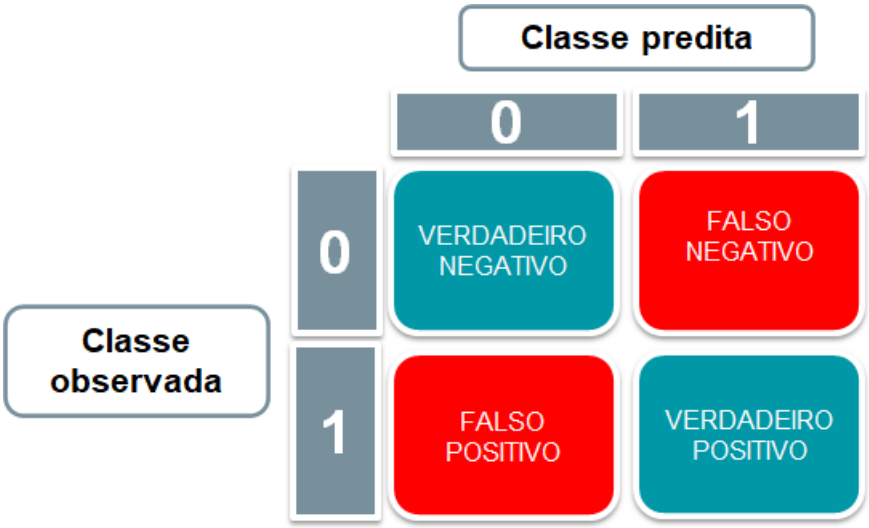
A eficiência de um modelo QSAR, muitas vezes, é medida a partir da comparação entre os valores reais e os previstos para a propriedade de interesse. A "matriz de confusão" (Figura 12), por exemplo, é uma medida muito utilizada na solução de problemas de classificação, podendo ser aplicada à classificação binária e também a problemas de classificação multiclasse (KULKARNI; CHONG; BATARSEH, 2020).

Essa matriz é uma tabulação cruzada dos rótulos observados: reais e os previstos. Os elementos diagonais da matriz de confusão indicam previsões corretas, enquanto os fora da diagonal representam previsões incorretas (JAMES *et al.*, 2017).

A saída "VN" significa Verdadeiro Negativo e indica o número de exemplos negativos classificados com precisão. Do mesmo modo, "VP" significa Verdadeiro Positivo, que indica

o número de exemplos positivos classificados com precisão. O termo “FP” mostra um valor falso positivo, isto é, o número de exemplos negativos reais classificados como positivos e “FN” significa um valor falso negativo que é o número de exemplos positivos reais classificados como negativos (KULKARNI; CHONG; BATARSEH, 2020).

Figura 12 – Representação básica de Matriz de Confusão. Fonte: Autoria própria.



As células destacadas em azul claro na Figura 12 representam os casos em que as classificações foram previstas corretamente, enquanto que os elementos fora dessa diagonal foram aqueles rotulados incorretamente pelo modelo.

Com os valores da matriz de confusão obtidos, outras métricas podem ser calculadas para medir o desempenho do modelo. Dentre elas, destacam-se:

- **acurácia**: é a proporção de instâncias que são classificadas corretamente entre todas as amostras do conjunto de dados (HORVATH; ALDAHDOOH, 2017). Dito de outra forma, se refere ao quão frequente o classificador está correto.

$$acuracia = \frac{total\,de\,acertos}{total\,de\,elementos\,da\,amostra} = \frac{VP + VN}{(VP + FN) + (VN + FP)} \tag{2.2}$$

  Para isso, considera-se:

  - VP (Verdadeiro Positivo). Exemplificando: se um indivíduo testou positivo para a COVID-19 e ele tem essa doença, então é chamado de verdadeiro positivo.
  - VN (Verdadeiro Negativo). Seguindo o mesmo exemplo anterior, se o resultado do teste para a COVID-19 for negativo e o indivíduo não tem essa doença, então é classificado como verdadeiro negativo.
  - FN (Falso Negativo). Se o resultado do teste da COVID-19 for negativo e o indivíduo estiver com a doença, então é chamado de falso negativo.

– FP (Falso Positivo). Se o resultado do teste para a COVID-19 for positivo para o indivíduo que não tem essa doença, então é chamado de falso positivo. A equação 2.2 apresenta o cálculo de acurácia.

- **precisão**: é a relação entre o verdadeiro positivo e o número total de positivos previstos. Portanto, é a porcentagem do conjunto classificado corretamente. Isto é, daqueles compostos que foram classificados como corretos, quantos efetivamente estavam corretos (MAZZOLARI; VISTOLI, 2015; HORVATH; ALDAHDOOH, 2017).

$$precisao = \frac{VP}{(VP + FP)} \tag{2.3}$$

- **sensibilidade** (*recall*): é a proporção de previsões positivas corretas em comparação ao total de positivos da amostra, isto é, a capacidade do modelo em identificar todas as instâncias de interesse (MAZZOLARI; VISTOLI, 2015). Nessa medida, os falsos negativos são considerados mais prejudiciais que os falsos positivos.

$$sensibilidade = \frac{verdadeiros\,positivos}{total\,de\,positivos\,da\,amostra} = \frac{VP}{(VP + FN)} \tag{2.4}$$

- ***f-score*** (*F-measure*): é obtido a partir de uma média ponderada entre a sensibilidade e a precisão. O resultado dessa média está no intervalo entre [0, 1]. Quanto mais próximo de 1, melhor será o desempenho do modelo.

$$f - measure = \frac{(1 + \beta * precisao * sensibilidade)}{\beta^2 * precisao + sensibilidade} = \frac{(1 + \beta) * \frac{VN}{VN+FP} * \frac{VP}{VP+FN}}{\beta^2 * \frac{VN}{VN+FP} + \frac{VP}{VP+FN}} \tag{2.5}$$

- **especificidade**: é a proporção de previsões negativas corretas em comparação ao número total de instâncias negativas (MAZZOLARI; VISTOLI, 2015).

$$especificidade = \frac{verdadeiros\,negativos}{total\,de\,negativos\,da\,amostra} = \frac{VN}{(VN + FN)} \tag{2.6}$$

- **coeficiente Kappa**: é uma medida responsável por medir o grau de concordância, ou discordância, entre o que foi previsto e observado na classificação, variando entre 0 e 1 (VIEIRA; SOUSA, 2010). A Tabela 2 ilustra o cálculo do coeficiente Kappa, tendo como ponto de partida um problema de duas classes.

Tabela 2 – Matriz de confusão para um problema de duas classes, sendo N = o número total de classes, C1 e C2 indicam os rótulos relacionados com as classes 1 e 2, respectivamente.

Rótulo previsto				
Rótulo correto		**C1**	**C2**	**Total**
	C1	a	b	$a + b = C1_{corr}$
	C2	c	d	$c + d = C2_{corr}$
	Total	$a + c = C1_{pred}$	$b + d = C2_{pred}$	N

O coeficiente Kappa é definido por:

$$Kappa = \frac{N * (a + e + i) - (C1_{corr} * C1_{pred} + C2_{corr} * C2_{pred})}{N^2 - (C1_{corr} * C1_{pred} + C2_{corr} * C2_{pred})} \tag{2.7}$$

podendo ser generalizado para as classes m:

$$Kappa = \frac{N \sum_{i=1}^{m} CM_{ii} - \sum_{i=1}^{m} Ci_{corr} Ci_{pred}}{N^2 - \sum_{i=1}^{m} Ci_{corr} Ci_{pred}} \tag{2.8}$$

em que CM_{ii} representam os elementos diagonais da matriz de confusão (TALLON-BALLESTEROS; RIQUELME, 2014). O resultado do coeficiente Kappa é interpretado conforme consta na Tabela 3.

Tabela 3 – Interpretação dos valores do coeficiente de Kappa

Valor Kappa	**>0,20**	**0,21 - 0,40**	**0,41 - 0,60**	**0,61 - 0,80**	**0,81 - 1,00**
Qualidade do classificador	ruim	fraca	boa	muito boa	excelente

- **área sob a curva** (AUC)[5]: é uma medida que permite melhor visualização do desempenho do modelo. O espaço ROC (*Receiver Operating Characteristic*) representa os *tradeoffs* relativos entre benefícios (verdadeiros positivos) e custos (falsos positivos). Quanto mais próximo de 1, melhor será o desempenho do modelo, mas quanto mais próximo da diagonal, é possível inferir uma previsão aleatória (em torno de 0,5) (FAWCETT, 2006).

2.3.4 Domínio de aplicabilidade

A definição do domínio de aplicabilidade (AD)[6] de um modelo é uma etapa fundamental para maximizar a qualidade do próprio modelo (BOBROWSKI *et al.*, 2020). Um modelo produzirá previsões confiáveis quando suas hipóteses forem válidas e previsões não confiáveis quando forem violadas. Portanto, é importante definir o espaço onde as previsões do modelo são confiáveis (BASKIN; KIREEVA; VARNEK, 2010; MAZZOLARI; VISTOLI, 2015). Assim, o objetivo do AD é avaliar a precisão da previsão (ou confiabilidade) do modelo de acordo com a avaliação das moléculas e sua relação com o "domínio" do modelo (BASKIN; KIREEVA; VARNEK, 2010).

Neste contexto, a análise de similaridade entre os compostos do conjunto de treinamento é considerada uma abordagem para determinação da estimativa do domínio de aplicabilidade. Um composto terá uma previsão confiável se for muito semelhante com aqueles utilizados pelo algoritmo na fase de aprendizagem (BOBROWSKI *et al.*, 2020).

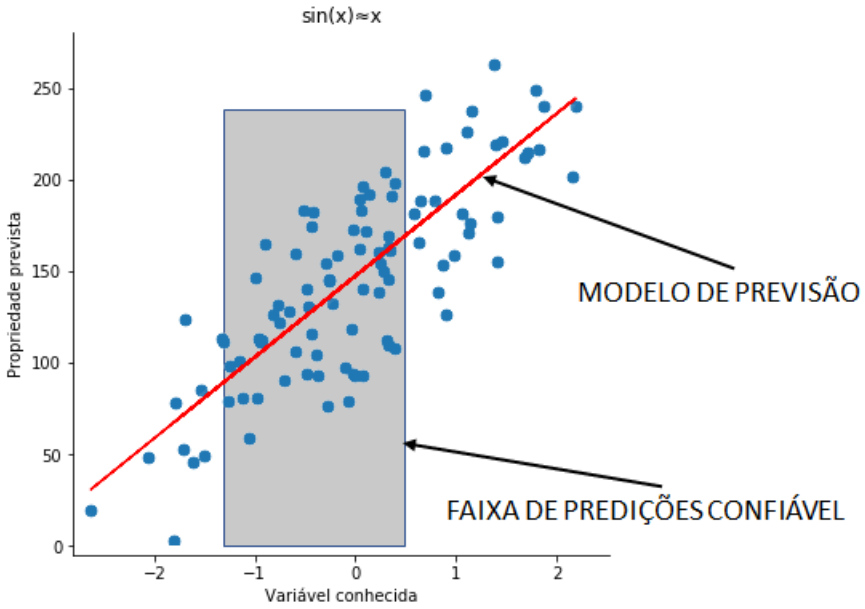
[5] Do inglês *Area Under The Curve*.
[6] Do inglês *Applicability Domain*.

A similaridade pode ser calculada em conformidade com critérios e o desempenho do modelo é impresso em relação a toda a gama de similaridade no conjunto de treinamento (MAZZOLARI; VISTOLI, 2015).

A confiabilidade não avaliada das previsões é o principal problema que restringe a aplicação prática dos modelos QSAR. Isto é, os modelos computacionais que têm uma boa precisão de predição para os compostos que foram usados para construir e validar o modelo não têm garantia de um desempenho igualmente bom para compostos diferentes (novos). Logo, não existe um modelo computacional universal que funcione igualmente bem em todo o espaço químico (SUSHKO, 2011).

Desta forma, a falha em especificar a área de aplicabilidade do modelo (subespaço químico), determinando onde o modelo é válido e é suscetível em fornecer previsões precisas, é o fator limitante para a aplicação prática de modelos computacionais. Portanto, o problema da incerteza na precisão e na confiabilidade das previsões é abordado em uma área emergente de pesquisa, que é o domínio de aplicabilidade (SUSHKO, 2011). A Figura 13 ilustra um exemplo do problema relacionado ao AD.

Figura 13 – Exemplo ilustrativo para o problema do domínio de aplicabilidade. Fonte: Autoria própria.



Na região cinza, os dados são aproximados em um modelo linear (linha vermelha). Porém, fora dessa região, a aproximação não é válida. Portanto, o domínio de aplicabilidade do modelo linear está definido na região cinza (intervalo [-1, 1]) (SUSHKO, 2011).

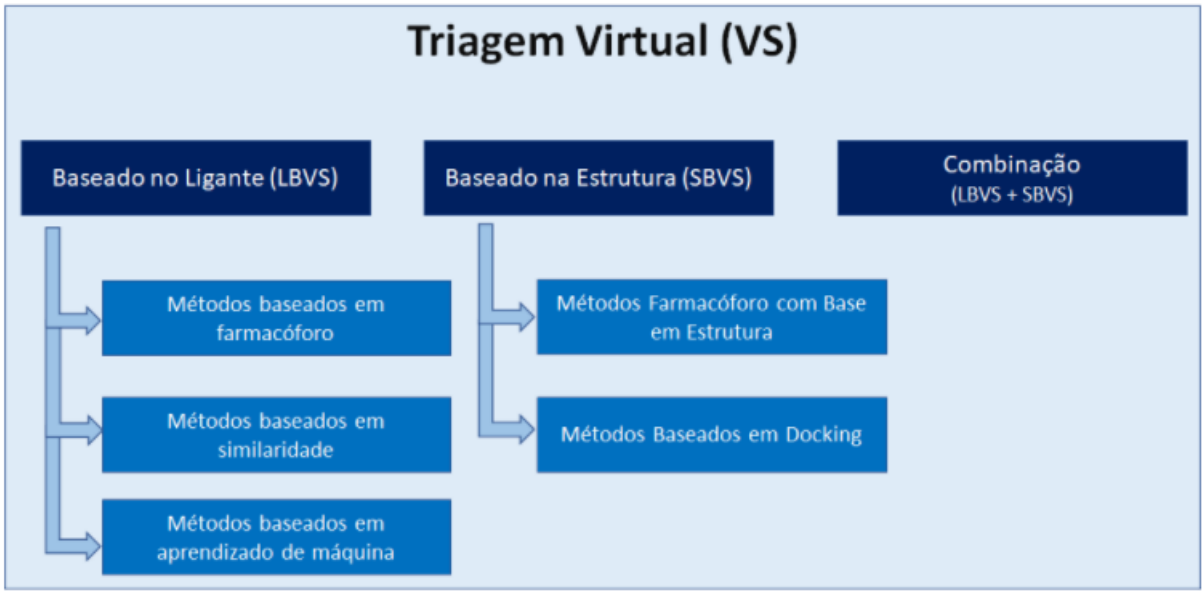
Após a descrição das etapas envolvidas na construção e validação dos modelos de QSAR, o próximo passo do trabalho envolve a aplicação destes modelos como filtros nos estudos de triagem virtual, a qual é uma das técnicas empregadas na identificação de novos candidatos a fármacos e que será descrita em mais detalhes a seguir.

2.4 Triagem Virtual

A triagem virtual é uma abordagem computacional usada para rastrear grandes bases de dados contendo moléculas pequenas em busca de substâncias com propriedades química/biológicas desejadas e que podem ser testadas experimentalmente (CARPENTER *et al.*, 2018; NEVES *et al.*, 2018).

A triagem virtual realiza as buscas por meio de simulação computacional (*in silico*) de centenas/milhares de compostos em estruturas de alvos biológicos, aumentando o rendimento e sucesso na descoberta de potenciais candidatos a fármacos. Desta forma, o aprendizado de máquina é uma poderosa ferramenta para auxiliar o processo de triagem virtual e, consequentemente, a descoberta de compostos como potenciais candidatos a fármacos (CARPENTER *et al.*, 2018; CARPENTER; HUANG, 2018). A estratégia de triagem virtual pode ser dividida em duas categorias (Figura 14): métodos baseados no ligante (LBVS)[7] e técnicas baseadas na estrutura do alvo (SBVS)[8] (KUMAR; KRISHNA; SIDDIQI, 2015).

Figura 14 – Diferentes abordagens empregadas em estudos de Triagem Virtual. Fonte: Autoria própria.



É importante ressaltar que a finalidade da triagem virtual "não é substituir ensaios in *vitro* ou *in vivo*, mas acelerar o processo de descoberta, reduzir o número de candidatos a serem testados experimentalmente e racionalizar sua escolha, proporcionando economia de tempo, custo, recursos e mão-de-obra" (NEVES *et al.*, 2018).

[7] Do inglês *Ligand-based Virtual Screening*.
[8] Do inglês *Structure-based Virtual Screening*.

Dentre as diferentes estratégias para execução da triagem virtual, modelos QSAR podem ser utilizados como filtros nas etapas iniciais da VS. Em geral, os modelos QSAR são usados para prever a propriedade biológica de novos compostos e podem ser considerados ferramentas valiosas devido ao seu alto e rápido rendimento, além de boa taxa de acerto (CARPENTER *et al.*, 2018; NEVES *et al.*, 2018).

Cabe ainda ressaltar que as triagens virtuais baseadas no aprendizado de máquina "estão entre as técnicas menos caras em termos de computação e tiveram um sucesso significativo" nas últimas décadas (CARPENTER *et al.*, 2018). Elas incluem "a seleção de um conjunto de compostos filtrados, constituídos por agentes ativos e inativos conhecidos. Após o treinamento do modelo, ele é validado e, se suficientemente preciso, usado em bancos de dados não vistos anteriormente podem ser usados para rastrear novos compostos com a desejada atividade frente ao alvo de interesse" (CARPENTER; HUANG, 2018).

Alguns autores (CARPENTER *et al.*, 2018) propõem o seguinte fluxo de trabalho ao empregar aprendizado de máquina em estudos de triagem virtual: '"uma vez construído e considerado satisfatório um modelo de aprendizado de máquina (modelo treinado e validado), ele pode ser usado para conduzir uma simulação VS em bibliotecas quimio-genômicas extremamente grandes. Os compostos com maior pontuação são chamados de *hits* (acertos) e estão sujeitos a testes *in vitro* para verificar se apresentam atividade biológica desejada. O rendimento destes testes é muito superior ao de uma triagem normal de alto rendimento, uma vez que o modelo obtido via aprendizado de máquina já previu a interação composto-alvo biológico. A partir deste ponto, os compostos mais promissores (chamados *leads* - derivações) podem ser desenvolvidos e testados, esperançosamente se tornando fármacos" (CARPENTER *et al.*, 2018).

Existem diversos algoritmos/classificadores que podem ser aplicados na triagem virtual. Alguns exemplos de aplicação do aprendizado de máquina na triagem virtual incluem:

(a) descoberta de fármacos para a doença de Alzheimer (CARPENTER; HUANG, 2018). Algoritmos de aprendizado de máquina usados: *Naïve Bayes; k-Nearest Neighbors; Support Vector Machines; Artificial Neural Networks; Ensemble Methods.*

(b) previsão de interação proteína-composto (CHEN *et al.*, 2018). Algoritmos de aprendizado de máquina usados: algoritmos baseados em similaridade (métodos do vizinho mais próximo, modelos locais bipartidos, métodos de fatoração da matriz); algoritmos baseados em vetores de características (Florestas Aleatórias).

(c) identificação de potenciais inibidores da proteína-tirosina fosfatase 1B (PTP1B) - um alvo terapêutico para diabetes tipo 2 e obesidade. Algoritmos de aprendizado de máquina utilizados: *naïve Bayesian, random forest, support vector machine* e *k-nearest neighbor* (CHAMJANGALI, 2020).

Além destes, redes neurais artificiais foram usadas para a previsão de inibidores de protease para o vírus HIV[9] (RAO *et al.*, 2009), para previsão da permeabilidade à barreira hematoencefálica e ligação à soroalbumina (KARELSON *et al.*, 2008), para previsão de inibidores da Furina, capazes de evitar a maturação das toxinas produzidas pelo *Bacillus anthracis*, (WORACHARTCHEEWAN *et al.*, 2009) e para a geração de um modelo de QSAR usado na previsão de toxicidade de *pirril-aril-sulfonas*, utilizadas como inibidores não nucleosídicos de transcriptase reversa para o tratamento da AIDS (CHAMJANGALI, 2020).

Considerando toda a contextualização e fundamentos teóricos abordados até aqui, a condução de uma triagem virtual se faz necessária para aumentar o número de potenciais candidatos a fármacos, por meio de algoritmos de aprendizado de máquina. Esse enfoque computacional representa um avanço fundamental na pesquisa da doença de Alzheimer, pois a demanda por novos tratamentos eficazes e preventivos é iminente, dada a crescente prevalência da doença em uma população envelhecida.

[9] Do inglês *Human Immunodeficiency Virus*

3 TRABALHOS RELACIONADOS

Nesta seção serão apresentados o estado arte sobre o tema de pesquisa, além de outras aplicações técnicas de modelos de aprendizado de máquina.

3.1 Estado da arte

Após realizar uma consulta na literatura, um conjunto de artigos foram encontrados, os quais fornecem diferentes abordagens para o desenvolvimento de modelos QSAR. Dentre elas, destacam-se as técnicas de aprendizado de máquina supervisionado e não supervisionado, aprendizado profundo, rede neural convolucional, técnicas de aprendizado de transferência e aprendizado de máquina baseado em grafos (SAKAI *et al.*, 2021; GUPTA *et al.*, 2021). Além disso, esses artigos exploram diferentes conjuntos de descritores moleculares para a predição da atividade de inibidores da AChE para a doença de Alzheimer (MOUCHLIS *et al.*, 2020).

A AChE é uma enzima que desempenha um papel importante para a degradação da acetilcolina no cérebro, afetando diretamente a função cognitiva. Por isso, o desenvolvimento de novos inibidores da AChE é uma área de interesse na pesquisa de novos fármacos para o tratamento da doença de Alzheimer (BAO *et al.*, 2023).

Alguns autores ressaltam que uma abordagem híbrida pode ser uma estratégia eficaz para a identificação de compostos naturais com atividade inibitória contra múltiplos alvos na doença de Alzheimer. Além disso, a combinação de abordagens de modelagem molecular e QSAR pode ser útil para a seleção e priorização de compostos para avaliação experimental adicional (DAS; CHAKRABORTY; BASUCORRESPONDING, 2019).

Neste estudo de Dhamodharan e Mohan (2022), foram desenvolvidos modelos de aprendizado de máquina para prever a eficácia de inibidores da AChE e BACE1 no tratamento da doença de Alzheimer. Foram usados diversos descritores moleculares e métodos de aprendizado, obtendo modelos estatisticamente significativos. Esses modelos podem ser usados no projeto de novos tratamentos para a doença de Alzheimer (DHAMODHARAN; MOHAN, 2022).

Após as leituras realizadas, percebeu-se que uma abordagem promissora para tratar a doença de Alzheimer é a inibição da AChE, pois ela é uma das principais proteínas envolvidas na degradação da acetilcolina, um neurotransmissor crucial para a função cognitiva e sua inibição pode, potencialmente, melhorar os sintomas da doença de Alzheimer, como a perda de memória e a deterioração cognitiva. A predição da atividade de inibidores da AChE é uma tarefa importante para o desenvolvimento de novos fármacos para a doença de Alzheimer (DAI *et al.*, 2022).

Além disso, uma técnica muito utilizada para a predição da atividade de compostos químicos é o uso de modelos QSAR, pois correlacionam a estrutura molecular de um composto com sua atividade biológica. O principal desafio é construir modelos QSAR precisos e confiáveis, tendo em vista a complexidade da relação entre a estrutura molecular e a atividade biológica. Nos últimos anos, houve um aumento considerável de modelos QSAR, combinando diferentes abordagens de aprendizado de máquina e descritores moleculares para melhorar a precisão das predições (BAO *et al.*, 2023; DAI *et al.*, 2022).

Por exemplo, alguns estudos combinaram a abordagem de aprendizado de máquina *Random Forest* com diferentes tipos de descritores moleculares para a predição da atividade de inibidores dda AChE e BACE1. Neste estudo (HU *et al.*, 2019), os autores concluíram que *Random Forest* é o melhor modelo de aprendizado para a previsão de drogas e alvos de Alzheimer.

Outra abordagem promissora para a construção de modelos QSAR é a utilização de redes neurais artificiais (ANNs - Artificial Neural Networks) combinadas com descritores moleculares. As ANNs são capazes de aprender relações complexas entre descritores moleculares e atividade biológica, e várias estratégias foram propostas para a construção de modelos QSAR baseados em ANNs (DOBCHEV; KARELSON, 2016; CHEIRDARIS, 2020; DHAMODHARAN; MOHAN, 2022).

Por último, destaca-se que a construção de modelos QSAR para a predição da atividade de inibidores da AChE para a doença de Alzheimer é um tema de pesquisa em constante evolução, e diferentes abordagens de aprendizado de máquina e descritores moleculares estão sendo explorados para melhorar a precisão das predições. Acredita-se que esses modelos sejam úteis para o desenvolvimento de novos fármacos para essa doença, contribuindo com o aceleramento do processo de descoberta de medicamentos e reduzindo os custos associados.

3.2 Abordagens utilizadas para o desenvolvimento de modelos QSAR

A abordagem de QSAR é amplamente utilizada para a descoberta de medicamentos e no desenvolvimento de fármacos para diversas doenças, incluindo a doença de Alzheimer. Para melhorar a precisão e robustez dos modelos QSAR na predição da atividade de inibidores da AChE para essa condição, abordagens combinadas têm sido exploradas, combinando diferentes métodos de aprendizado de máquina e descritores moleculares. Essas abordagens objetivam aproveitar as vantagens de cada componente para fornecer resultados mais confiáveis e abrangentes (TODESCHINI; CONSONNI, 2000; GOLBRAIKH; TROPSHA, 2003).

A combinação de diferentes tipos de descritores moleculares é uma das abordagens mais comuns usadas no desenvolvimento de modelos QSAR. Os descritores moleculares são

representações numéricas que capturam características estruturais das moléculas, como informações topológicas, físico-químicas e de conectividade. Ao combinar descritores 2D e 3D, por exemplo, torna-se possível contemplar uma variedade de informações moleculares importantes para a atividade inibitória da AChE. Essa abordagem permite melhorar a capacidade preditiva dos modelos (FARA; A.L.; OPREA, 2019).

Outra estratégia adotada em modelos QSAR é a junção de diferentes algoritmos de aprendizado de máquina e aprendizado profundo. Esses algoritmos podem incluir regressão linear, redes neurais, métodos de aprendizado profundo e outros. Cada algoritmo possui suas próprias capacidades e limitações na captura de relações complexas entre os descritores moleculares e a atividade inibitória da AChE. Ao combinar esses algoritmos em um modelo, é possível aproveitar suas vantagens individuais e obter uma previsão mais precisa e confiável (CARPENTER; HUANG, 2018).

Além disso, a validação cruzada e a avaliação de desempenho realizadas de modo adequado são importantes no desenvolvimento de modelos QSAR. A divisão adequada dos conjuntos de treinamento e teste, juntamente com técnicas como validação externa e bootstrapping, permitem avaliar a robustez e a generalização dos modelos. Essas etapas são fundamentais para garantir que os modelos QSAR sejam confiáveis e possam fornecer previsões precisas e úteis (PANOV; DZEROSKI, 2007; PARVANDEH *et al.*, 2020).

Essas abordagens têm o potencial de impulsionar a descoberta e o desenvolvimento de novos compostos terapêuticos com maior eficácia e se mostram promissoras na luta contra essa doença neurodegenerativa.

3.3 Outras aplicações

Dada a rápida disseminação da COVID-19 e sua alta mortalidade, torna-se urgente descobrir medicamentos específicos para combater o vírus SARS-CoV-2 (GUY *et al.*, 2020). Nesse contexto, técnicas de aprendizado de máquina têm sido utilizadas para apoiar a triagem virtual na busca por inibidores de alvos moleculares relacionados com SARS-CoV-2. Uma dessas proteínas, a protease M^{pro}, é essencial dentro do ciclo viral, ou seja, seus inibidores poderiam bloquear a replicação viral (TEJERA *et al.*, 2020).

Uma busca na literatura revelou algumas aplicações de técnicas de aprendizado de máquina no contexto da COVID-19. Dentre elas, destaca-se o uso de aprendizado de máquina em imagens médicas para diagnosticar pneumonia relacionada à COVID-19. Vale ressaltar que os modelos construídos e suas avaliações tiveram um alto risco de viés, ocasionados em virtude de relatórios e uma combinação inadequada de pacientes com e sem COVID-19. No entanto, nenhum dos 145 modelos de previsão construídos foram recomendados para serem usados na prática (WYNANTS *et al.*, 2020).

Em outro estudo, ferramentas computacionais de biologia estrutural e aprendizado

de máquina (rede neural artificial) foram utilizadas para prever a presença de antígenos e identificar potenciais epítopos de células T. epítopo ou determinante antigênico é a menor porção de antígeno com potencial de gerar a resposta imune. O referido estudo também fez uso de algoritmo de acoplamento (*docking*) computacional para estimar a superfície de SARS-CoV-2 que interage com seu receptor humano conhecido (ACE2) (FAST; CHEN, 2020).

Algoritmos baseados no aprendizado de máquina também foram utilizados para melhorar identificação de casos de COVID-19, usando uma pesquisa na web baseada em telefone celular, capturando as manifestações mais comuns da doença (sinais e sintomas), juntamente com o histórico básico de viagens dos usuários (RAO; VAZQUEZ, 2020).

Por último, a combinação do algoritmo de aprendizado de máquina supervisionado (árvores de decisão) com processamento digital de sinais foi usada para realizar análises de genoma. O método proposto identifica uma assinatura genômica do vírus responsável pela COVID-19 e a usa, em conjunto com uma abordagem livre de alinhamento baseada no aprendizado de máquina, para uma classificação ultrarrápida, escalonável e altamente precisa de genomas inteiros do vírus SARS-CoV-2 (RANDHAWA *et al.*, 2020).

4 PROPOSTA DE SOLUÇÃO: TRIAGEM VIRTUAL UTILIZANDO CONSENSO ENTRE MODELOS QSAR E BUSCA POR SIMILARIDADE

Neste trabalho, uma pesquisa aplicada foi realizada, pois "objetiva gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos" (GIL, 2017). Além disso, foi utilizada uma abordagem mista quanti-qualitativa, ou seja, foram realizadas pesquisas que combinam elementos de abordagens de pesquisa qualitativa e quantitativa com o propósito de ampliar e aprofundar o entendimento sobre os temas de pesquisa e a confirmação/validação dos resultados (JOHNSON; ONWUEGBUZIE; TURNER, 2007).

A pesquisa também tem um caráter exploratório, pois "proporciona maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a constituir hipóteses, tendo como principal objetivo o aprimoramento de ideias ou a descoberta de intuições" (GIL, 2017). Desta forma, as seguintes etapas serão realizadas ao longo deste trabalho:

- pesquisa bibliográfica, visando compreender os conceitos necessários para realização do estudo, assim como identificar na literatura os trabalhos correlatos e refinar a metodologia proposta.

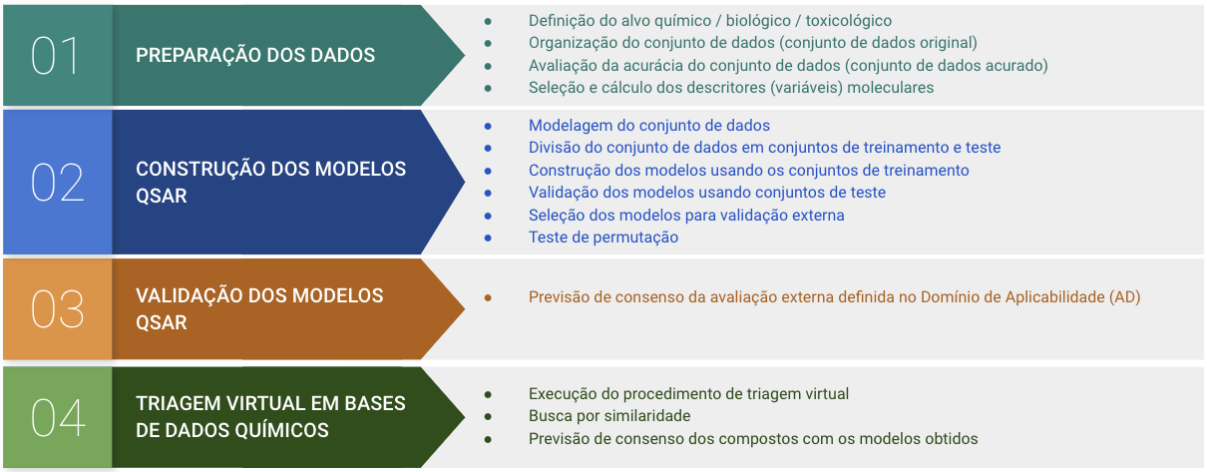
- pesquisa experimental, realizando uma investigação empírica na qual o pesquisador manipula e controla variáveis independentes e observa as variações que tal manipulação e controle produzem em variáveis dependentes. Variável é um valor que pode ser dado por quantidade, qualidade, característica, magnitude, variando em cada caso em particular. Variável independente é aquela que influencia, determina ou afeta a dependente. Variável dependente é aquela que vai ser afetada pela independente.

- estudo de caso, a ser realizado empregando bases de dados contendo substâncias químicas e dados biológicos, assim como algoritmos de aprendizado de máquina e aprendizado profundo a fim de identificar potenciais candidatos a fármacos para o tratamento da doença de Alzheimer, assim como avaliar o desempenho de diversas técnicas de aprendizado de máquina.

4.1 Estruturação da metodologia empregada

A metodologia proposta para realização deste trabalho foi estruturada em quatro etapas, como ilustra a Figura 15.

Figura 15 – Principais etapas envolvidas na metodologia empregada neste trabalho. Fonte: Adaptado de (TROPSHA *et al.*, 2017)



4.1.1 Etapa 01 - Preparação dos dados

4.1.1.1 Definição do alvo químico / biológico / molecular

O alvo biológico definido foi a enzima Acetilcolinesterase, também conhecida como AChE. A AChE é uma enzima envolvida na degradação do neurotransmissor acetilcolina, desempenhando um papel fundamental na transmissão de sinais nervosos no sistema nervoso (DHAMODHARAN; MOHAN, 2022).

4.1.1.2 Organização do conjunto de dados (conjunto de dados original)

Para a construção dos modelos (conjunto de treinamento e testes, e a validação externa) foram utilizadas as seguintes bases de dados:

- ChEMBL (www.ebi.ac.uk/chembl), base utilizada para seleção das amostras (compostos químicos).
- DUD-E, dude.docking.org/targets/aces), base utilizada para obtenção de compostos de referência para a similaridade.
- PubChem (pubchem.ncbi.nlm.nih.gov/rest/pug), base utilizada para selecionar compostos para a etapa de triagem virtual.

4.1.1.3 Avaliação da acurácia do conjunto de dados (conjunto de dados acurado)

Para garantir a acurácia dos dados, adotou-se o fluxo proposto por Tropsha e colaboradores (FOURCHES; MURATOV; TROPSHA, 2016):

- passo 1: Preparo, de um ponto de vista químico, do conjunto de dados, que segue um protocolo previamente estabelecido e permite a identificação e correção de erros nas estruturas químicas (FOURCHES; MURATOV; TROPSHA, 2010).
- passo 2: Duplicatas identificadas (compostos repetidos) são analisadas e removidas.
- passo 3: Realiza-se uma análise da variabilidade experimental intra e interlaboratorial.
- passo 4: Exclusão de fontes de dados não confiáveis, ou seja, dados com alta variação nos valores dos ensaios.
- passo 5: Detecção e análise dos "*cliffs*" relacionados aos dados de atividade biológica (MAGGIORA, 2006).

Todas as estruturas químicas e informações biológicas correspondentes foram padronizadas usando o Standardizer v.20.8.0 (ChemAxon, Budapest, Hungary, disponível em: www.chemaxon.com) (ALVES *et al.*, 2021). A partir desta ferramenta, compostos inorgânicos, contra-íons, metais, compostos organometálicos e misturas foram removidos. Além disso, quimiotipos específicos, como anéis aromáticos e grupos nitro, foram normalizados. Também foram excluídas as duplicatas da seguinte forma: (i) se as duplicatas tivessem atividade biológica diferente, ambas as entradas foram excluídas; e (ii) se os resultados relatados para as duplicatas fossem os mesmos, uma entrada era mantida no conjunto de dados e a outra era excluída (ALVES *et al.*, 2021).

4.1.1.4 Cálculo dos descritores (variáveis/atributos) moleculares

Três tipos de estratégias computacionais foram utilizadas para a geração de descritores 2D para as amostras (compostos) do conjunto de dados:

- função *Fingerprints* de Harry Morgan (FIGUERAS, 1993);
- *software* para geração de descritores SiRMS (*Simplex Representation of Molecular Structure*), disponível em www.qsar4u.com/pages/sirms.php
- biblioteca RDKit (rdkit.Chem.MoleculeDescriptors.MolecularDescriptorCalculator), disponível em www.rdkit.org/docs/source/rdkit.ML.Descriptors.MoleculeDescriptors.html.

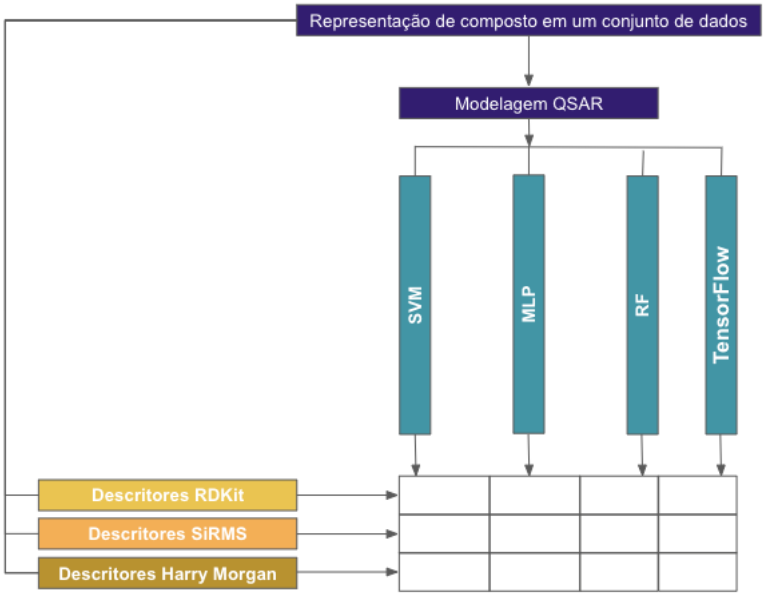
4.1.2 Etapa 02 - Construção dos modelos QSAR

4.1.2.1 Conjuntos de dados

Os conjuntos de dados utilizados neste trabalho foram compostos de descritores gerados por diferentes ferramentas (Morgan, SiRMS e RDKit). O bloco Y (variável dependente) é formado por dados biológicos de uma coleção de compostos, ativos e inativos, enquanto que o bloco X (variáveis independentes) é composto por um conjunto

de descritores moleculares referentes a cada estratégia de obtenção. A Figura 16 ilustra a combinação entre os algoritmos de aprendizado de máquina e os tipos de descritores selecionados para análise.

Figura 16 – Representação esquemática ilustrando o conjunto de dados usado para construção dos modelos (SVM, MLP, RF e TensorFlow), combinados com os descritores calculados (RDKit, SiRMS e Morgan). Fonte: Autoria própria.



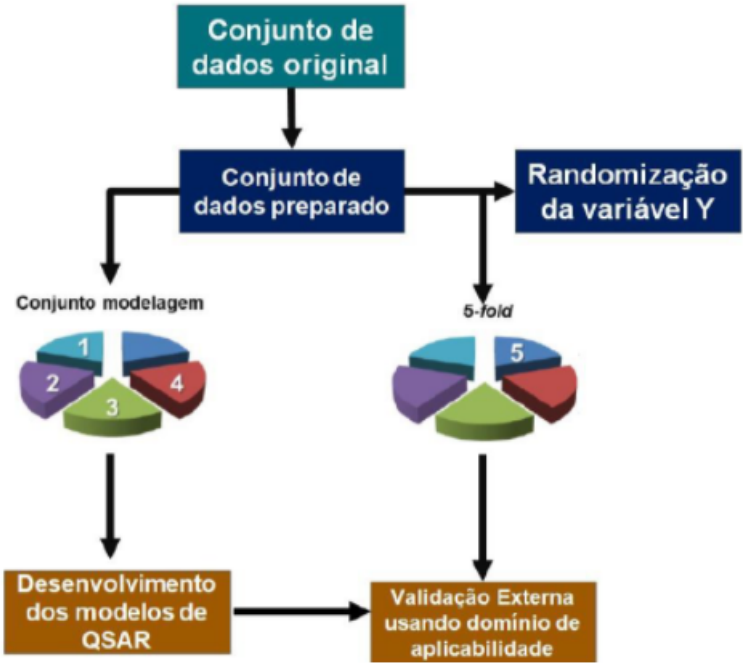
4.1.2.2 Divisão do conjunto de dados em conjuntos de treinamento e teste

A divisão dos conjuntos de dados (treinamento e testes) e de avaliação externa foi realizada selecionando aleatoriamente as instâncias e aplicando o método de validação cruzada 5-*fold* (TROPSHA *et al.*, 2017), levando em consideração as seguintes etapas:

- o conjunto de dados total com atividade experimental definida foi aleatoriamente dividido empregando a técnica de *bootstrap* em cinco subgrupos de tamanhos iguais;
- em seguida, um destes subgrupos (20% de todos os compostos) foi definido como conjunto de validação externa;
- os quatro conjuntos restantes formaram o conjunto de treinamento (80% de todo o conjunto de dados);
- esse procedimento foi repetido cinco vezes, permitindo que cada um dos cinco subconjuntos fosse usado como conjunto de validação externa;
- é importante ressaltar que o conjunto de validação externa nunca foi usado na construção e/ou seleção dos modelos;
- na validação externa, através dos 20% dos dados do conjunto original, foi avaliado o desempenho dos modelos treinados e testados. Para tanto, foi utilizado a validação

cruzada estratificada com a mesma parametrização utilizada para o treinamento e testes, conforme ilustrado na Figura 17.

Figura 17 – Etapas empregadas no desenvolvimento dos modelos de aprendizado de máquina. Fonte: Autoria própria.



4.1.2.3 Construção dos modelos usando os conjuntos de treinamento

O esquema geral utilizado para construção dos modelos combinou três algoritmos supervisionados (SVM, MLP e RF) e um algoritmo de aprendizado profundo utilizando a biblioteca *TensorFlow*, com os quatro tipos de descritores moleculares obtidos nas etapas prévias (Figura 16). Os modelos foram construídos usando a linguagem *Python* 3 e as seguintes bibliotecas foram utilizadas (PEDREGOSA *et al.*, 2011):

- sklearn.ensemble.RandomForestClassifier (para RF), disponível em: scikit-learn.org;
- scikit-learn 0.23.2 (para SVM), disponível em: scikit-learn.org.

Nesta etapa, foi utilizada uma técnica para otimização de hiperparâmetros para fins de comparação: a Busca aleatória: `sklearn.model_selection.RandomizedSearchCV`, disponível em: scikit-learn.org.

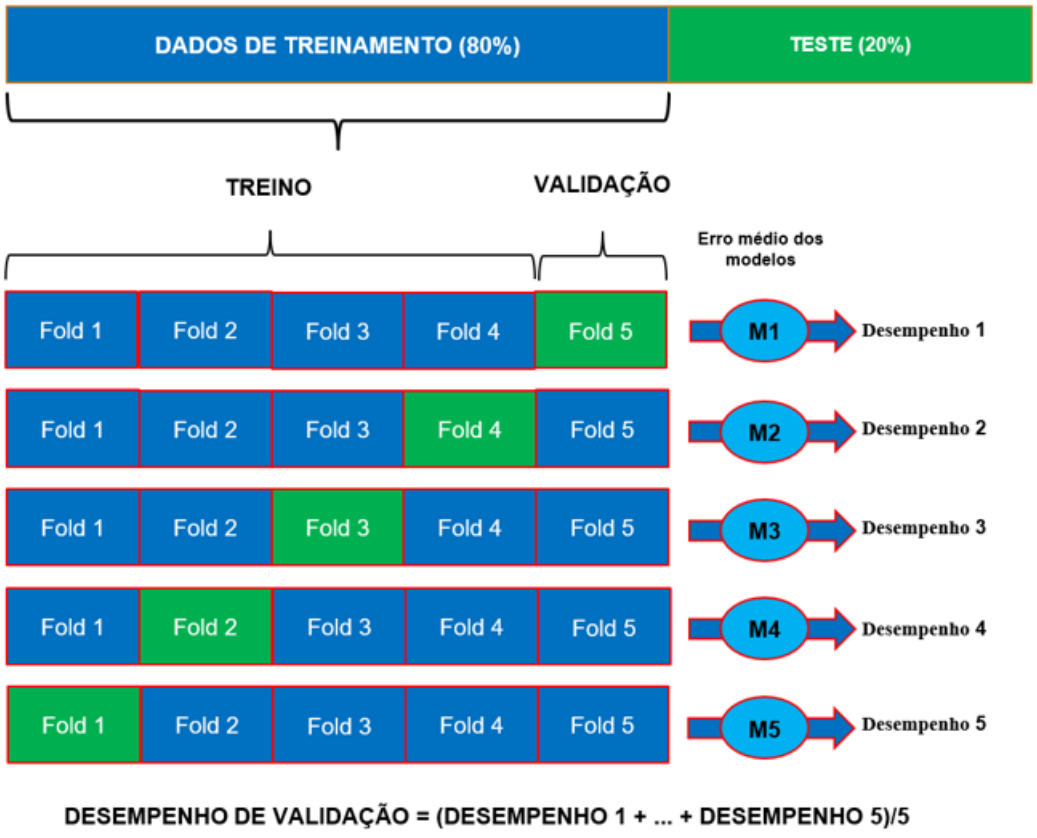
4.1.2.4 Validação dos modelos usando conjuntos de teste

Para avaliação do poder de generalização dos modelos, a técnica de validação cruzada (CV) 5-*fold* foi utilizada com base na biblioteca `sklearn.model_selection` com o método *StratifiedKFold* e o parâmetro de número de divisão igual a 5.

O conjunto de dados total com atividade experimental definida foi dividido em cinco subgrupos de tamanhos iguais. Então, um destes subgrupos (20% de todos os compostos) foi definido como conjunto de validação externa e os quatro conjuntos restantes formaram o conjunto de treinamento (80% de todo o conjunto de dados).

Esse procedimento foi repetido cinco vezes, permitindo que cada um dos cinco subconjuntos fosse usado como conjunto de validação externa. Os modelos foram gerados usando apenas o conjunto de treinamento. É importante enfatizar que o conjunto de validação externa nunca foi empregado para geração e/ou seleção dos modelos. Cada conjunto de modelagem é dividido em vários conjuntos de treinamento e teste; então os modelos são gerados usando compostos de cada conjunto de treinamento e aplicados aos conjuntos-teste para avaliar a robustez e a capacidade preditiva dos modelos. A Figura 18 ilustra o processo para execução da validação cruzada.

Figura 18 – Processo empregado na etapa de validação cruzada. Fonte: Autoria própria.



4.1.2.5 Seleção dos modelos para validação externa

Nessa etapa, os modelos foram avaliados de acordo com as seguintes métricas:

- acurácia (ACC);
- sensibilidade (Se) e especificidade (Sp);

- valor preditivo positivo (VPP);
- valor preditivo negativo (VPN);
- área sob a curva ROC (AUC);
- medida F ou F *score*;
- coeficiente Kappa de Cohen (*Cohen's k*);

4.1.2.6 Teste de permutação

O teste de permutação tem o objetivo de avaliar se o modelo sofreu *overfitting* (sobreajuste). Para tanto foi empregada a função *permutation_test_score* (*sklearn.model_selection*) em scikit-learn (PEDREGOSA *et al.*, 2011), com 10 permutações e a validação cruzada de 5 *folds*, conforme recomendação de (JORNER *et al.*, 2021).

O valor de p foi avaliado com a finalidade de indicar a completa falta de aprendizado, quando há randomização dos dados. Isso permite a busca por uma forte evidência de que os modelos não estão apenas aprendendo ruído, mas estão encontrando um valor real.

4.1.3 Etapa 03 - Validação dos modelos

4.1.3.1 Previsão de consenso da avaliação externa definida no Domínio de Aplicabilidade

O domínio de aplicabilidade foi definido a partir das seguintes etapas:

1. avaliação da similaridade molecular: a similaridade molecular de um *bit* de impressão digital corresponde a um fragmento da molécula obtido a partir da impressão digital das moléculas, tendo como métrica a similaridade padrão obtida pelo coeficiente de Tanimoto. Nesta etapa, foi utilizado o programa KNIME *Analytics* (KNIME, 2021), com a biblioteca RD*Kit* e a função *rdkit.DataStructs.FingerprintSimilarity()*. Vale destacar que cada *bit* de impressão digital corresponde a um fragmento da molécula onde as moléculas semelhantes têm muitos fragmentos em comum.

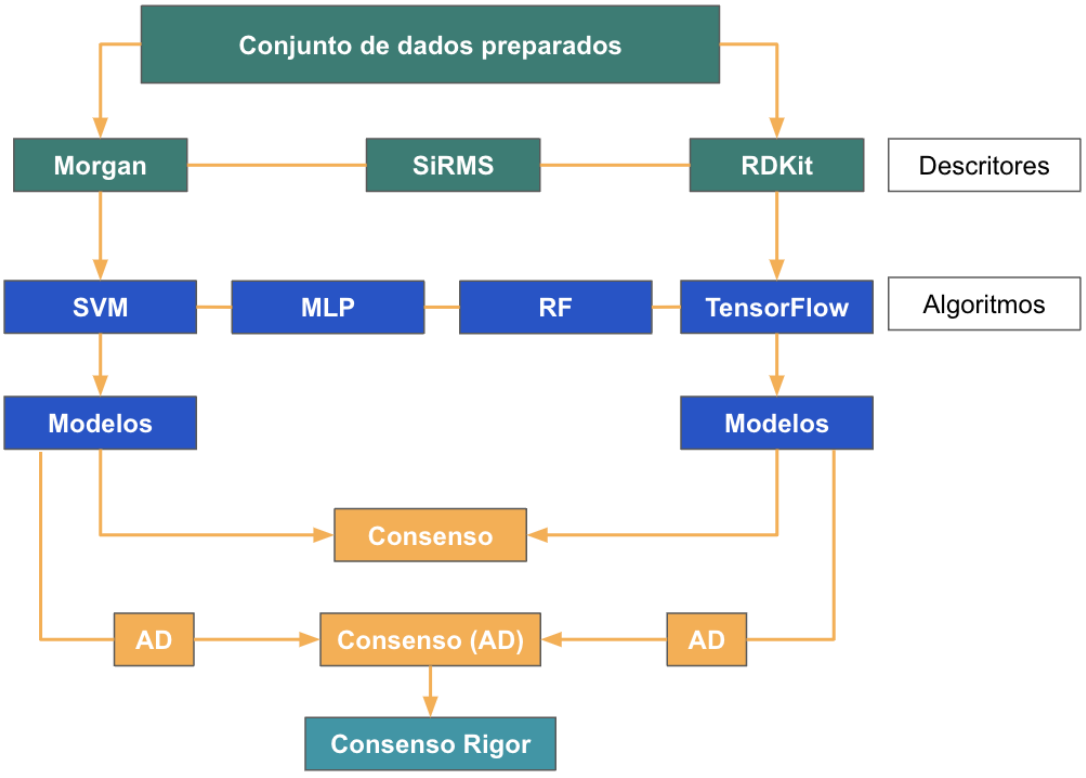
2. depois da avaliação de similaridade molecular, o valor da probabilidade associada à previsão de cada instância dentro do grupo de moléculas similares é definida, variando de acordo com cada algoritmo de classificação. A função *predict_proba* foi utilizada para a obtenção da força de ligação a um rótulo ou *score* (variável *threshold_ad*) de cada instância a um rótulo (0 ou 1 - ativo ou inativo, respectivamente) calculados para cada algoritmo.

3. após o cálculo do valor de AD, modelos que apresentaram *score* maior que o limite AD (*threshold_ad*) foram classificados como modelos AD.

Após a identificação do grupo de moléculas que formam o AD, foi medido o grau de confiança das previsões (o quão certo um modelo de aprendizado de máquina está sobre sua previsão) de cada molécula a um rótulo (Figura 19). Isto foi considerado da seguinte forma, após avaliação se a molécula recebeu um valor previsto (ativo ou inativo), por descritor:

1. se a molécula obteve consenso de um mesmo rótulo (ativo ou inativo) em todos os descritores, esta molécula se enquadra no grupo de **Consenso**.
2. se essa molécula está também no grupo AD de todos os descritores (*score* maior que o limite AD calculado), ela se enquadra no grupo de **AD** do respectivo descritor em questão.
3. se a molécula está em todos ADs, de todos os descritores, com o mesmo rótulo, ela se enquadra no grupo de **Consenso AD**.
4. enfim, o valor do rótulo referente à molécula é comparado em todos os descritores, descritores AD e consenso AD. Se o rótulo é o mesmo, a molécula fará parte do grupo **Consenso Rigor**.

Figura 19 – Etapas empregadas para a previsão de consenso e construção do AD. Fonte: Autoria própria.



A Tabela 4 ilustra um exemplo de como tabular os dados.

Tabela 4 – Exemplo de tabulação de consenso para obtenção do AD.

SMILES	Descritor 1	Desc. 1 AD	Desc. 2	Desc. 2 AD	Consenso	Consenso AD	Consenso Rigor
Molécula 1	1	1	1	1	1	1	1
Molécula 2	0	0	0		0		

4.1.4 Etapa 04 - Triagem virtual em bases de dados químicos

4.1.4.1 Previsão de consenso dos compostos com os modelos obtidos

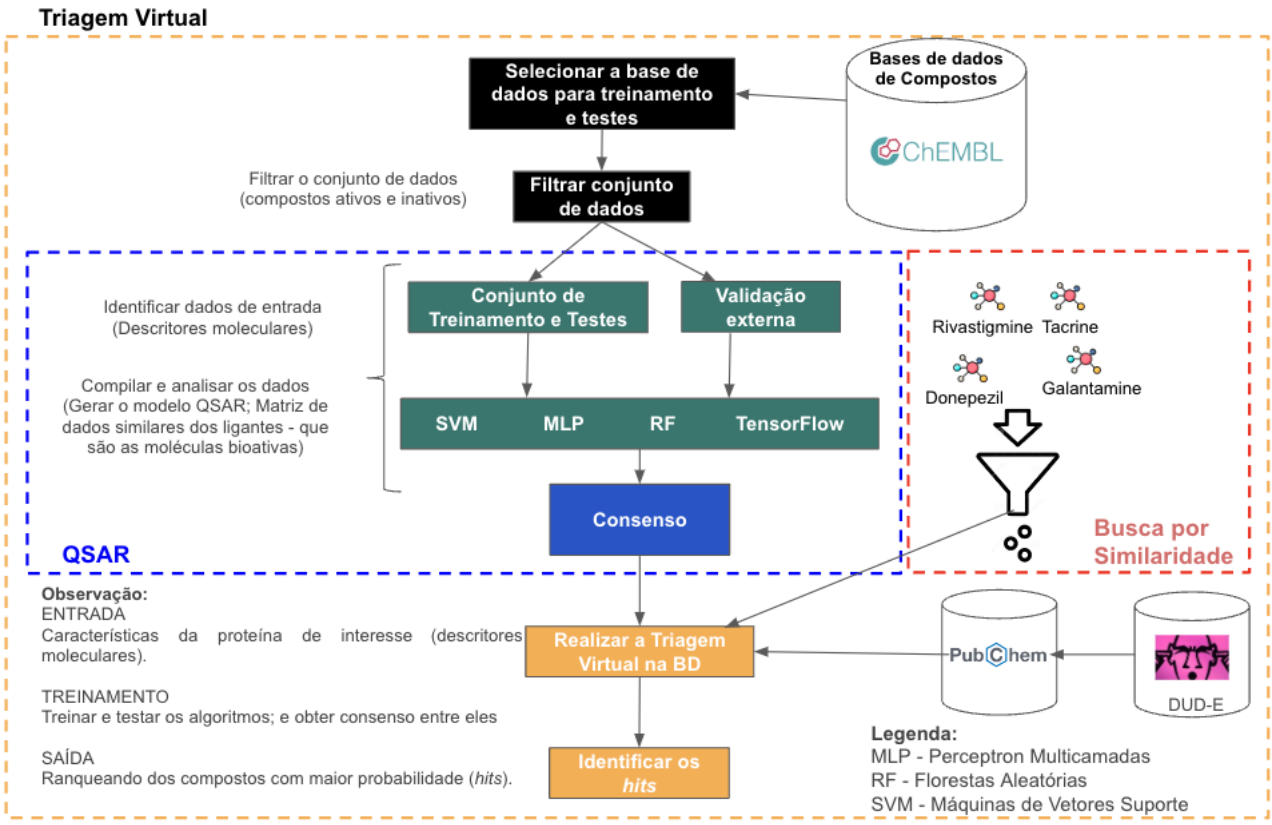
Os modelos do grupo "Consenso com rigor" obtidos a partir do consenso dos descritores demonstram quais modelos (obtidos a partir de diferentes tipos de descritores e diferentes algoritmos) são mais eficientes na previsão da propriedade-alvo (atividade biológica). A seguir, os modelos podem ser aplicados em uma grande base de dados químicos como filtros moleculares, onde novamente é avaliada a capacidade preditiva de cada modelo.

O procedimento de consenso (Etapa 03) dos resultados dos modelos para cada molécula é também aplicado durante a triagem virtual. As moléculas com maior força de ligação dentro do consenso (*hits*) ao rótulo de "Ativo" são selecionadas.

4.1.4.2 Execução do procedimento de triagem virtual

A Figura 20 apresenta o passo-a-passo para execução da triagem virtual, etapa que será realizada futuramente.

Figura 20 – Método proposto para realização da triagem virtual. Fonte: Autoria própria.



A próxima seção apresenta os resultados obtidos após a realização de todas as etapas previstas neste estudo.

5 AVALIAÇÃO EXPERIMENTAL

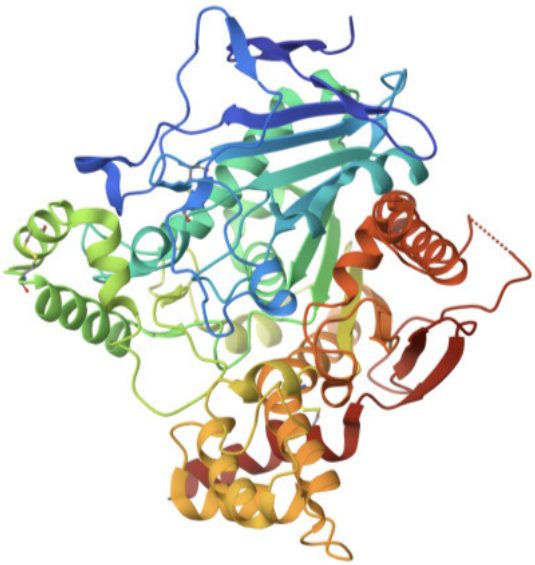
Os resultados alcançados neste trabalho serão apresentados e discutidos neste capítulo, que se destinam a construção e validação de modelos QSAR direcionados à doença de Alzheimer. Estes modelos foram utilizados como filtros moleculares em uma grande base de dados de compostos com a finalidade de identificar candidatos a fármacos como potenciais inibidores da enzima AChE. Os resultados completos deste estudo estão disponíveis no GitHub para acesso público.

5.1 Preparação dos dados

5.1.1 Definição do alvo químico

O alvo biológico definido foi a enzima Acetilcolinesterase (AChE) (Figura 21) em função da sua importância na fisiopatologia da doença de Alzheimer, pois ela está envolvida na degradação da acetilcolina e sua inibição pode aliviar os sintomas da doença. Além disso, a AChE é um biomarcador reconhecido da doença (WALCZAK-NOWICKA; HERBET, 2021).

Figura 21 – Estruturas de acetilcolinesterase. Fonte: www.rcsb.org/structure/1b41



A construção do conjunto de dados de treinamento e teste teve início com um total de 8.832 compostos químicos, os quais foram submetidos a testes de inibição da AChE, usando a API do banco de dados ChemBL (*(chembl-webresource-client)*) (CHEMBL, 2023).

5.1.2 Organização e avaliação da acurácia do conjunto de dados original

O conjunto de dados foi estruturado em duas categorias distintas: uma destinada aos dados de treinamento e teste, e a outra voltada para a triagem virtual. Os detalhes sobre cada uma delas serão fornecidos nas seções subsequentes.

5.1.2.1 Dados de treinamento e testes

Para garantir que os compostos utilizados para o treinamento atendam aos requisitos necessários para serem considerados candidatos a fármacos viáveis, as cinco regras de Lipinski (LIPINSKI *et al.*, 1997) foram aplicadas:

- massa molecular inferior a 500 Daltons;
- não mais que 5 doadores de ligações de hidrogênio;
- não mais que 10 aceitadores de ligações de hidrogênio;
- coeficiente de partição octan-1-ol/água (Log P) não superior a 5.

Após a conclusão da Análise Exploratória de Dados (EDA) usando os descritores e as cinco regras de Lipinski (RDKIT, 2023), 8.832 compostos permaneceram na nossa amostra.

Após a etapa de seleção de recursos, foi necessário realizar uma leve correção no modelo de classificação, a qual consistiu na remoção de compostos químicos que não se classificavam nas categorias de ativos ou inativos. Essa ação simplifica a tarefa de classificação, uma vez que o modelo se concentra na previsão de apenas duas classes, representadas numericamente como 1 (ativo) ou 0 (inativo). As seguintes etapas foram executadas:

- **inicialização dos dados**: Inicia-se com um conjunto de dados com mais de 8.000 compostos químicos que foram testados para inibir a proteína acetilcolinesterase (AChE).
- **filtragem por proteína**: Os dados foram filtrados para selecionar apenas a proteína de interesse, a AChE.
- **filtragem por atividade padrão (IC50)**: Os dados foram novamente filtrados para manter apenas aqueles com atividade padrão do tipo IC50, que mede a concentração inibitória em 50%.
- **remoção de valores ausentes**: As linhas com valores ausentes na coluna "*standard_value*" foram removidas, resultando em 7.549 linhas.

- **remoção de valores ausentes**: As linhas com valores ausentes na coluna "*standard_value*" foram removidas, resultando em 7.549 linhas.

- **definição de classes de bioatividade**: Os valores na coluna "*standard_value*" foram convertidos em classes de bioatividade com base em limites. Compostos com valores maiores ou iguais a 10.000 foram considerados "inativos", aqueles com valores menores que 1.000 foram classificados como "ativos", e os demais como "*min effect*". Essa ação resultou em 3.570 compostos ativos, 2.187 inativos e 1.792 com "*min effect*".

- **criação de *DataFrame* com colunas selecionadas**: Um novo *DataFrame* ("data2") foi criado contendo apenas as colunas relevantes, incluindo a classe de bioatividade, o identificador de molécula, a estrutura química (SMILES) e o valor padrão. O *DataFrame* resultante teve 7.549 linhas.

- **cálculo de descritores de Lipinski**: Os descritores de Lipinski, que são características físicas das moléculas, foram calculados para os compostos usando a função "*mol_descriptors*". Esses descritores incluíram o peso molecular, o número de doadores de hidrogênio, o número de aceitadores de hidrogênio e o coeficiente de partição octanol-água (logP).

- **filtragem de *outliers* de pIC50**: Os valores de pIC50 (potência da atividade) foram normalizados e os *outliers* identificados e removidos, resultando em 7.487 compostos.

- **filtragem final de *outliers* de pIC50**: Os *outliers* foram novamente identificados e removidos, resultando em 7.483 compostos.

- **remoção de linhas com valores ausentes**: As linhas com valores ausentes foram removidas do *DataFrame*, mantendo 7.483 compostos.

- **remoção de coluna redundante**: A coluna "*standard_value*" foi removida do *DataFrame*.

- **filtragem de compostos que violam as regras de Lipinski**: As regras de Lipinski (Peso molecular < 500 daltons, logP < 5, número de doadores de hidrogênio < 5, número de aceitadores de hidrogênio < 10) foram verificadas para cada composto. Os compostos que violaram mais de uma regra foram excluídos, resultando em 6.385 compostos.

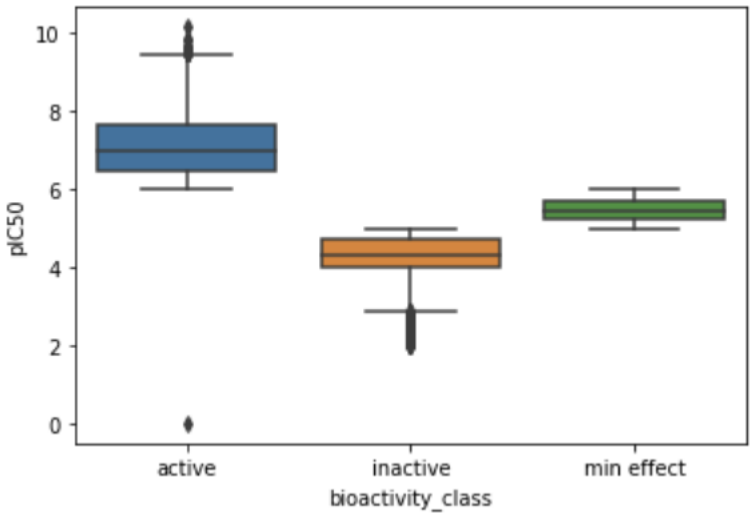
- **salvando dados para classificação e regressão**: Os dados foram separados em dois *DataFrames*: um para classificação (com coluna "*bioactivity_class*") e outro para regressão (sem coluna "*bioactivity_class*").

- **preparação para a engenharia de características**: Os dados foram preparados para a engenharia de características, selecionando as colunas "*canonical_smiles*" e "*molecule_chembl_id*", salvando-as em arquivos separados.
- **resultado final**: O *DataFrame* resultante contém 4.829 compostos químicos após todas as etapas de filtragem e preparação de dados.

O resultado final consiste em um conjunto de dados preparado e pronto para ser usado na criação de modelos de aprendizado de máquina, tanto para as tarefas de classificação quanto para regressão.

A Figura 22 ilustra o intervalo interquartil (IQR) da distribuição dos dados, categorizados em três classes distintas com base no pIC50:

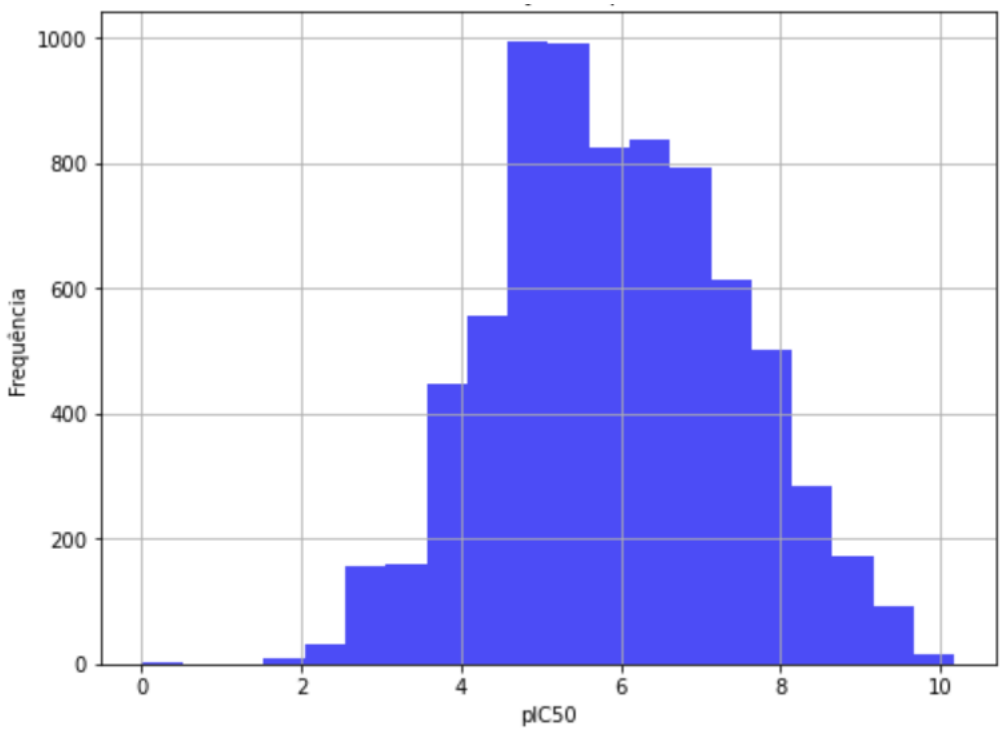
Figura 22 – Distribuição do pIC50 - Alvo (candidatos válidos). Fonte: Autoria própria.



- classe "*active*": Composta por 3.507 amostras, esta classe possui uma média de pIC50 de, aproximadamente, 7.24, com uma dispersão moderada, indicada por um desvio padrão de cerca de 0.89. Os valores variam de 0 (mínimo) a 10.19 (máximo), sendo a maioria dos dados concentrados entre 6.52 (primeiro quartil) e 7.80 (terceiro quartil).
- classe "*inactive*": Composta por 2.186 amostras, essa classe apresenta uma média de pIC50 de cerca de 4.22 e um desvio padrão de, aproximadamente, 0.67. Os valores dessa classe variam de 2.00 (mínimo) a 5.00 (máximo), com a maior parte dos dados situada entre 3.98 (primeiro quartil) e 4.75 (terceiro quartil).
- classe "*min effect*": Com 1.790 amostras, esta classe apresenta uma média de pIC50 em torno de 5.48, com baixa dispersão indicada por um desvio padrão de cerca de 0.29. Os valores dessa classe variam de 5.00 (mínimo) a 6.00 (máximo), sendo a maioria dos dados concentrados entre 5.23 (primeiro quartil) e 5.72 (terceiro quartil).

A Figura 23 apresenta a distribuição da frequência dos dados com base no pIC50, apresentando alguns pontos importantes a serem considerados:

Figura 23 – Distribuição da frequência de pIC50. Fonte: Autoria própria.



- **diversidade de dados**: O conjunto de dados parece ser rico e diversificado, abrangendo uma gama de valores de pIC50 para cada classe. Essa diversidade é fundamental para treinar modelos robustos que possam generalizar bem para novos dados, contribuindo para evitar o *overfitting*.

- **número de amostras**: O número total de amostras para cada classe ("*active*", "*inactive*" e "*min effect*") é razoável, o que é importante para treinar modelos estatisticamente significativos. No entanto, vale ressaltar que quanto mais dados, geralmente é melhor, especialmente para os modelos de aprendizado profundo.

- **diferenças significativas entre classes**: As estatísticas mostram que as médias de pIC50 são significativamente diferentes entre as classes, sendo um indicativo positivo, pois sugere que os modelos têm potencial para aprender a distinguir entre as diferentes classes de atividade biológica.

- **baixa dispersão em "*min effect*"**: A classe "*min effect*" apresenta uma baixa dispersão, indicada por um desvio padrão baixo, em comparação com as outras duas classes. Isso pode ser um desafio, pois os modelos podem ter dificuldade em distinguir as amostras dessa classe devido à sua proximidade nas características.

- **dados desbalanceados**: Destaca-se que o número de amostras em cada classe está desbalanceado, com a classe "*active*" tendo mais amostras do que as outras duas. Isso pode exigir técnicas de balanceamento de dados durante o treinamento do modelo para evitar qualquer viés resultante do desequilíbrio das classes.

5.1.2.2 Dados para triagem virtual

Nesta seção é apresentado o fluxo para a criação de um conjunto de dados (*dataset*) para a triagem virtual de compostos químicos, usando o PubChem como fonte de informações. A triagem virtual é um processo computacional essencial para a descoberta de medicamentos, cujo objetivo é identificar compostos químicos que têm potencial para se ligar a uma proteína alvo específica, nesse caso, a AChE (CARPENTER; HUANG, 2018). Os seguintes passos desse fluxo foram executados:

- passo 1: leitura de ligantes conhecidos em um arquivo
  - Neste primeiro passo, ligantes conhecidos para a Acetilcolinesterase (AChE) foram extraídos de um arquivo chamado "*actives_final.ism*". Esses ligantes foram usados como consultas-chave na triagem virtual.
  - **Total de ligantes conhecidos:** 453 ligantes.
- passo 2: busca de similaridade no PubChem
  - Neste segundo passo, cada um dos ligantes conhecidos foi usado como consulta em uma busca de similaridade no PubChem. O objetivo foi encontrar compostos químicos disponíveis no PubChem que compartilhem semelhanças estruturais com os ligantes conhecidos.
  - **Total de compostos semelhantes encontrados:** 159.470.
- passo 3: exclusão dos compostos de consulta dos resultados
  - Como vários ligantes conhecidos foram usados como consulta, existiu a possibilidade de alguns deles fossem devolvidos como resultados da busca de similaridade, usando outros ligantes como consulta. Para evitar duplicações, neste terceiro passo, os compostos de consulta foram excluídos dos resultados, deixando apenas os compostos não duplicados.
  - **Total de compostos após a exclusão:** 158.597.
- passo 4: filtragem de compostos não adequados para medicamentos
  - Neste quarto passo, os compostos químicos nos resultados da busca foram filtrados com base em quatro propriedades moleculares: número de doadores de ligação de hidrogênio, número de receptores de ligação de hidrogênio, peso

molecular e logP (partição octanol-água). Esses critérios estão alinhados aos critérios estabelecidos na regra dos cinco de Lipinski, os quais visam contribuir com a identificação de compostos que têm maior probabilidade de se tornarem medicamentos:

  * número de doadores de ligações de hidrogênio (*HBondDonorCount*): os compostos com até 5 doadores de ligações de hidrogênio foram mantidos. Número de compostos que atenderam a este critério: 158.360
  * número de Receptores de Ligações de Hidrogênio (*HBondAcceptorCount*): os compostos com até 10 receptores de ligações de hidrogênio foram mantidos. Número de compostos que atenderam a este critério: 157.829
  * peso Molecular (*MolecularWeight*): os compostos com um peso molecular igual ou inferior a 500 foram mantidos. Número de compostos que atenderam a este critério: 146.837
  * coeficiente de Partição Octanol-Água, XLogP (LogP): os compostos com um valor de LogP menor que 5 foram mantidos. Número de compostos que atenderam a este critério: 119.056

- Finalmente, o *DataFrame* foi filtrado para reter apenas os compostos que atenderam a todos os critérios de Lipinski, simultaneamente, resultando em um total de 117.379 compostos adequados para experimentos de triagem virtual ou ancoragem molecular.

- passo 5: desenho das estruturas dos 10 principais compostos
  - Neste quinto passo, as estruturas químicas dos 10 principais compostos da base de dados acurada foram desenhadas e exibidas, baseada na similaridade (CID).

- passo 6: extração de compostos exclusivos com base em SMILES canônicos
  - Neste sexto passo, os compostos foram submetidos a um filtragem para garantir que apenas as estruturas únicas fossem mantidas, com base em seus SMILES canônicos, ajudando a reduzir a redundância na lista de compostos.
  - **Compostos únicos após filtragem:** 117.379

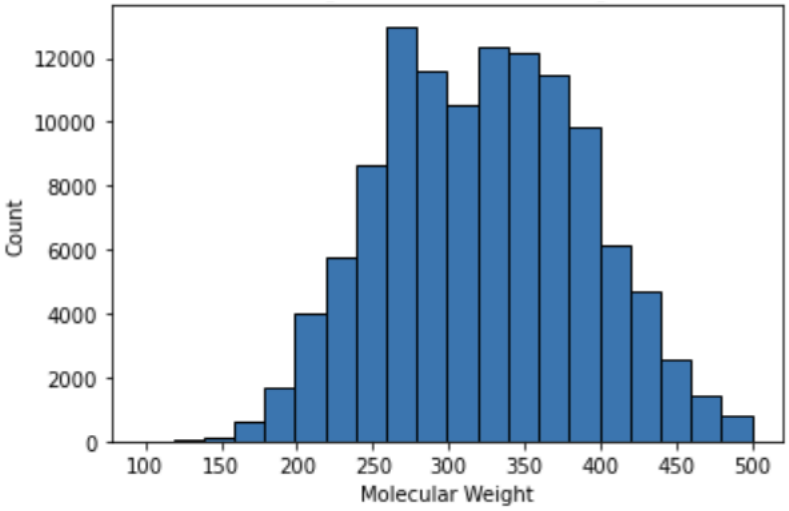
- passo 7: salvando os compostos em arquivos
  - Por fim, os compostos químicos resultantes foram salvos em arquivos no formato .mol, preparando-os para serem utilizados em experimentos de ancoragem molecular ou triagem virtual.

Vale ressaltar que, à medida que os critérios de triagem foram sendo aplicados, o número de compostos filtrados foi diminuindo, resultando em um conjunto final de

compostos candidatos que atendem aos requisitos de propriedades moleculares desejadas para potenciais medicamentos. Os dados também revelam informações sobre a dispersão de propriedades, como peso molecular e LogP, nos compostos que atendem aos critérios de Lipinski (LIPINSKI *et al.*, 1997).

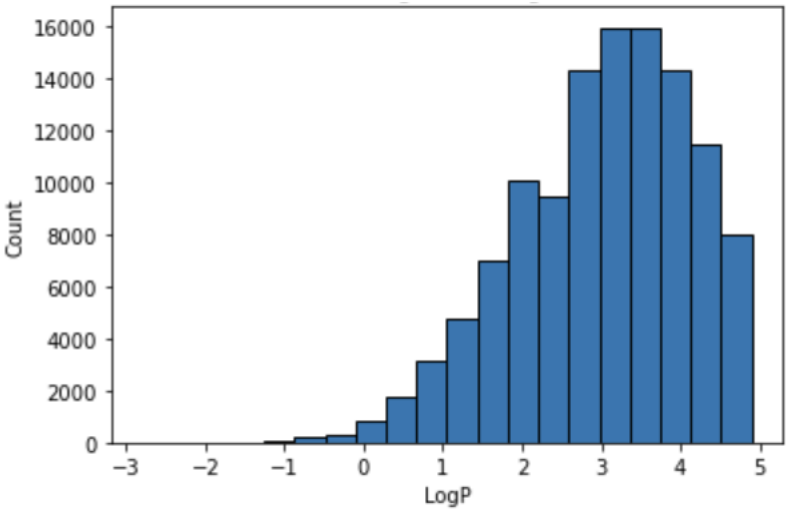
A Figura 24 ilustra o histograma de peso molecular, revelando que a maioria das observações se concentra no intervalo entre 298.985 e 319.0665, totalizando 12.346 observações. Esse gráfico permite observar como as contagens de observações variam à medida que o peso molecular aumenta ou diminui.

Figura 24 – Histograma do peso molecular. Fonte: Autoria própria.



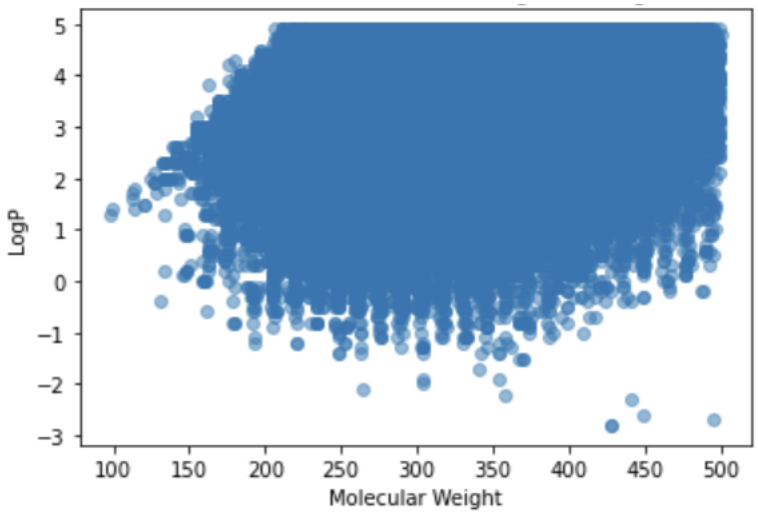
A Figura 25 representa o histograma LogP. Os dados revelam que a maioria das observações tem um LogP entre 1.05 e 1.435, totalizando 4.740 observações nesse intervalo. Além disso, o gráfico permite observar como a contagem de observações varia conforme o valor de LogP aumenta ou diminui.

Figura 25 – Histograma do LogP. Fonte: Autoria própria.



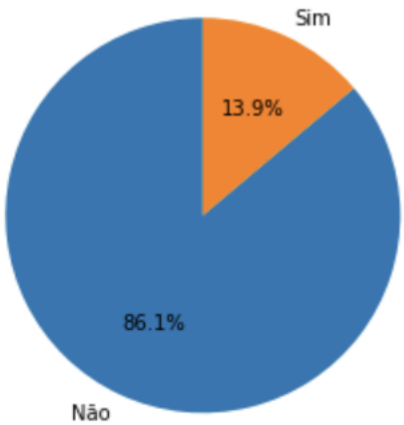
A dispersão dos pontos no gráfico (Figura 26) demonstra como as duas variáveis *MolecularWeight* e XLogP estão relacionadas. Esse gráfico é útil para identificar as tendências, padrões ou correlações entre as variáveis. A concentração dos pontos em uma área específica indica a existência de uma possível correlação ou relação entre as duas variáveis.

Figura 26 – Gráfico de dispersão de peso molecular vs LogP. Fonte: Autoria própria.



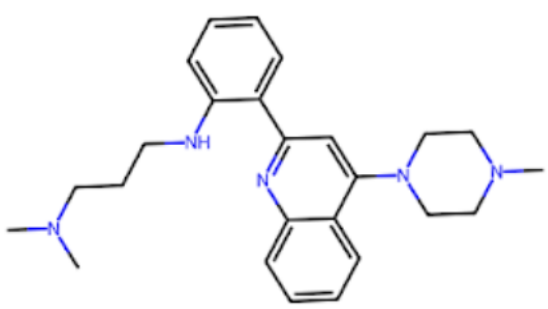
Portanto, ao término do processo, foram incluídos um total de 117.379 compostos (Figura 27) para a realização da triagem virtual.

Figura 27 – Distribuição do percentual de compostos incluídos e não incluídos. Fonte: Autoria própria.

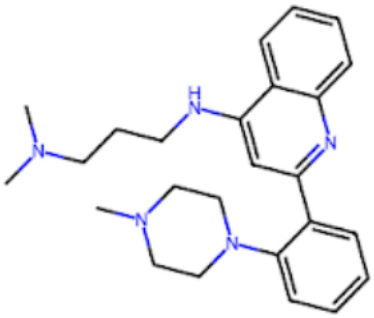


As Figuras (28, 29, 30, 31, 32) ilustram os 10 principais compostos da base de dados acurada para a realização da triagem virtual.

Figura 28 – Compostos 1 e 2. Fonte: Autoria própria.

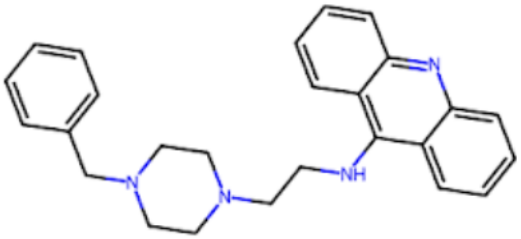


CID 16096265

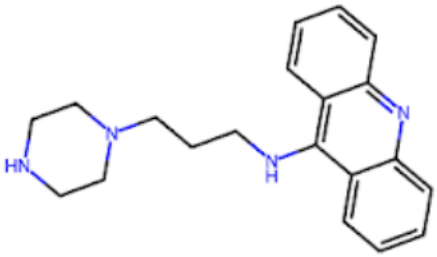


CID 16096263

Figura 29 – Compostos 3 e 4. Fonte: Autoria própria.

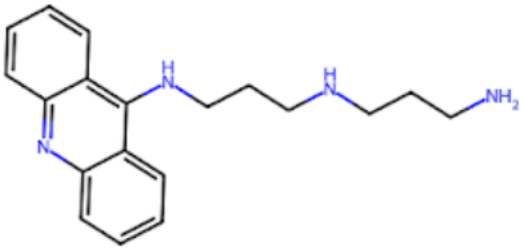


CID 44531417

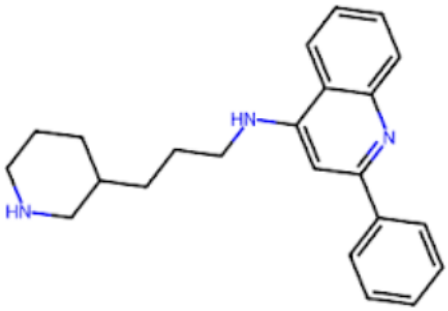


CID 163672604

Figura 30 – Compostos 5 e 6. Fonte: Autoria própria.



CID 145740211



CID 130365871

Figura 31 – Compostos 7 e 8. Fonte: Autoria própria.

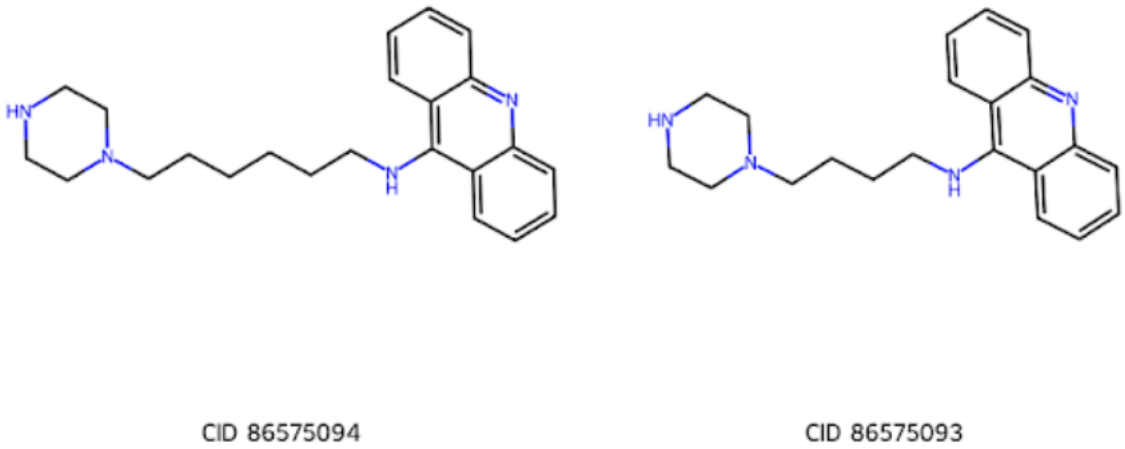
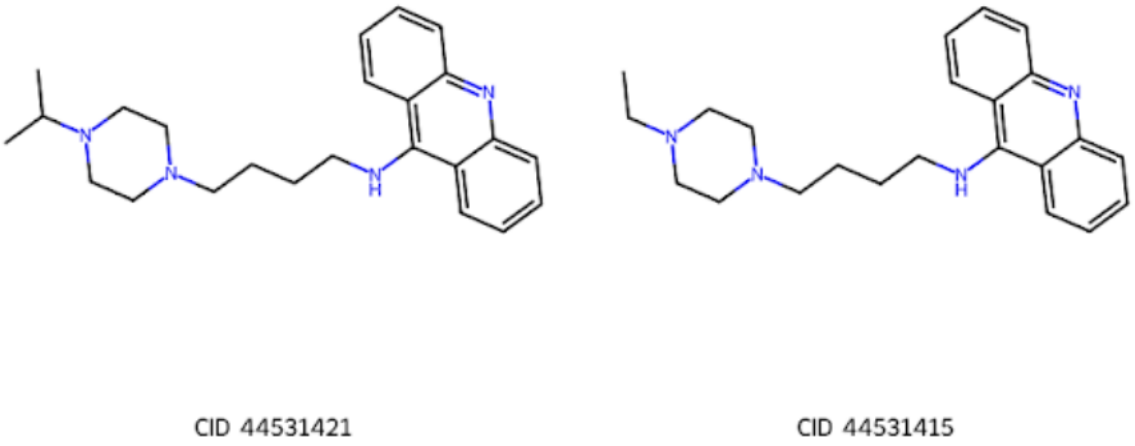


Figura 32 – Compostos 9 e 10. Fonte: Autoria própria.



5.1.3 Seleção e cálculo dos descritores (variáveis) moleculares

Neste estudo, ferramentas e métodos foram utilizados para a seleção e cálculo de descritores moleculares, com o objetivo de caracterizar as moléculas químicas da base de dados. Os principais resultados incluem:

- **RDKit e *MolecularDescriptorCalculator***: a biblioteca RDKit, uma ferramenta de química computacional em Python, foi utilizada para calcular vários descritores moleculares. Esses descritores numéricos resumem as características fundamentais das moléculas, tais como: o tamanho, a forma e a polaridade. Dentre os descritores calculados, destacam-se o LogP (coeficiente de partição octanol-água) e o peso molecular. Esses descritores são muito utilizados em química medicinal e modelagem molecular (RASHDAN; ABDELMONSEF, 2022).
- **Impressões Digitais Moleculares (*Morgan Fingerprint*)**: o RDKit também proporcionou a capacidade de calcular impressões digitais moleculares, especifica-

mente a Impressão Digital de Morgan. Essa representação binária (0s e 1s) da estrutura molecular é calculada pela função *GetMorganFingerprintAsBitVect*. A Impressão Digital de Morgan é baseada em *hashes* dos ambientes atômicos da molécula e é valiosa para tarefas de busca química e similaridade molecular, sendo frequentemente utilizada em triagem virtual de compostos químicos e química computacional (FIGUERAS, 1993).

- ***Simplex Representation of Molecular Structure (SiRMS)***: o SiRMS, ou Representação Simples da Estrutura Molecular, é um método que captura a topologia e a geometria das moléculas. Ele descreve moléculas como conjuntos de simplexos (polígonos tridimensionais) que refletem a conectividade entre átomos e a distância entre eles. Essa representação é aplicada em análises estruturais de moléculas e em simulações de dinâmica molecular, contribuindo para a compreensão detalhada das propriedades moleculares (XUE; BAJORATH, 2000).

5.2 Construção dos modelos QSAR

5.2.1 Modelagem do conjunto de dados

Neste estudo, realizou-se a modelagem do conjunto de dados visando a preparação dos mesmos para a criação e validação de modelos de aprendizado de máquina e aprendizado profundo. O conjunto de dados final, após todas as etapas de seleção e filtragem, foi composto de 4.829 amostras, distribuídas nas seguintes classes:

- **classe 1 (Ativo)**: este grupo é composto por 2.841 amostras. Essas amostras representam compostos químicos que demonstraram atividade como inibidores da enzima alvo (AChE) e são potenciais candidatos a fármacos para o tratamento da doença de Alzheimer.

- **classe 0 (Inativo)**: a classe de inativos é formada por 1.988 amostras. Essas amostras representam compostos que não demonstraram atividade significativa como inibidores da enzima AChE.

A distribuição dessas classes pode ser visualizada nas Figuras 33 e 34, que apresentam a frequência e densidade das amostras em cada classe.

Figura 33 – Frequência dos dados. Fonte: Autoria própria.

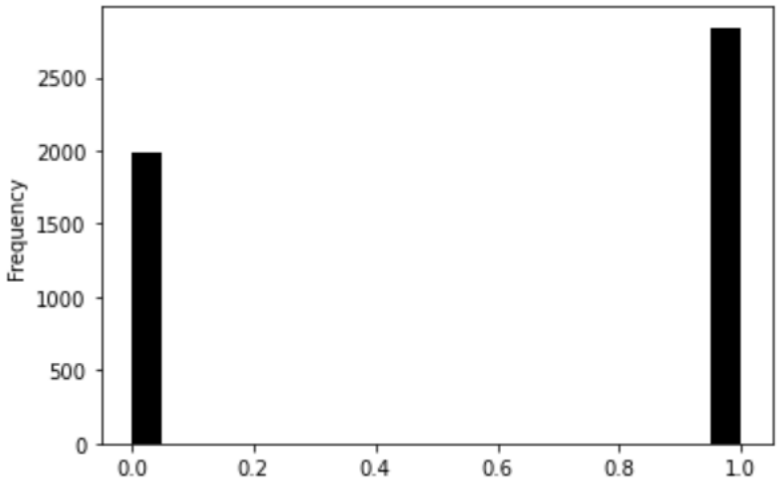
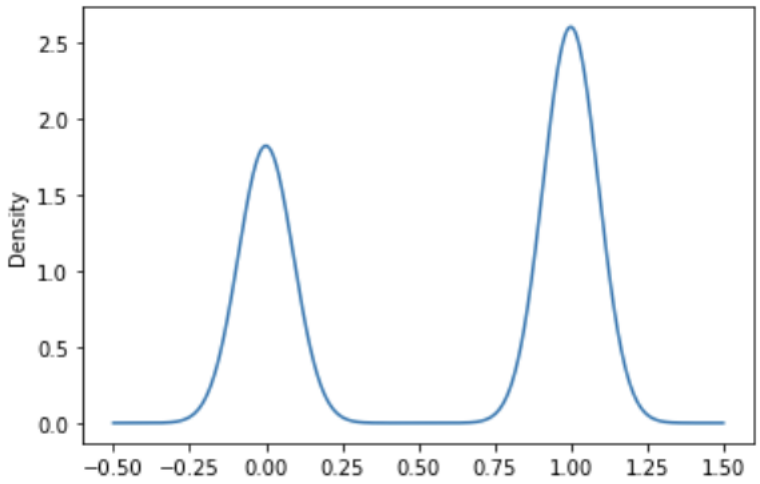


Figura 34 – Densidade dos dados. Fonte: Autoria própria.



Além disso, os descritores moleculares foram calculados, resultando nas seguintes características:

- **número de Entradas (Amostras)**: o conjunto de dados é composto por 4.829 entradas, representando as diferentes amostras químicas consideradas neste estudo.
- **colunas do Conjunto de Dados**: o conjunto de dados possui um total de 20 colunas, cada uma com informações específicas. Essas colunas incluem informações como identificadores, propriedades químicas, classes de bioatividade, dentre outras.
- **descritores Moleculares**: três conjuntos distintos de descritores moleculares foram calculados para cada amostra:
  - **Morgan**: 2048 descritores foram calculados usando o método Morgan, fornecendo informações detalhadas sobre a estrutura molecular.

- **RDKit**: 207 descritores foram obtidos utilizando a biblioteca RDKit, fornecendo informações adicionais sobre as características das moléculas.
- **SiRMS**: 1384 descritores foram gerados por meio do método SiRMS, capturando os aspectos topológicos e geométricos das moléculas.

5.2.2 Divisão do conjunto de dados em conjuntos de treinamento e teste

Na etapa de preparação do conjunto de dados, os dados foram divididos em conjuntos de treinamento e teste para permitir a avaliação do desempenho dos modelos de aprendizado de máquina e aprendizado profundo. Esses conjuntos foram assim divididos:

- **conjuntos de treinamento (*X_train* e *y_train*)**: esses conjuntos representam 80% dos dados originais. O conjunto *X_train* possui a forma (3.863, número de descritores), o que significa que contém 3.863 exemplos de treinamento, sendo que cada exemplo é caracterizado por um conjunto de descritores moleculares. Por sua vez, o conjunto *y_train* tem 3.863 descritores e contém os rótulos correspondentes para os exemplos de treinamento. Esses rótulos indicam a classe de bioatividade de cada amostra, ou seja, se o composto é ativo ou inativo em relação à enzima AChE.
- **conjuntos de validação externa (*X_val_ext* e *y_val_ext*)**: esses conjuntos são destinados à validação externa e representam os 20% restantes dos dados originais. O conjunto *X_val_ext* consiste de 966 exemplos de teste, cada um com descritores moleculares. O conjunto *y_val_ext*, por sua vez, tem 966 descritores e contém os rótulos correspondentes para os exemplos de teste, indicando suas classes de bioatividade.

5.2.3 Construção dos modelos usando os conjuntos de treinamento

Neste estágio do processo, esforços foram dedicados para a criação de modelos de aprendizado de máquina e aprendizado profundo, fazendo uma divisão de dados para treinamento e testes, validação externa, bem como uma busca por hiperparâmetros usando *RandomizedSearchCV*. Para tanto, foi feito:

- **divisão dos dados**: primeiramente, uma divisão estratégica dos dados foi feita nos conjuntos de treinamento e teste, bem como uma divisão adicional para a validação externa.
  - *X_train* e *y_train* compreendem 80% dos dados originais e foram usados para treinar os modelos.
  - *X_val_ext* e *y_val_ext* constituem os 20% restantes e foram reservados para a validação externa.

  - A divisão foi realizada usando a função *train_test_split*, onde a parcela de teste corresponde a 20% dos dados.

- **definição da validação cruzada estratificada**: é uma técnica crucial para a avaliação do desempenho do modelo, a qual foi adotada a abordagem de validação cruzada estratificada (*StratifiedKFold*) com 5 divisões. Essa estratégia garante que todos os subconjuntos incluam a mesma porcentagem de amostras positivas e negativas, tornando a avaliação mais confiável e justa (ALAMRO *et al.*, 2023).

- **busca por hiperparâmetros (RandomizedSearchCV)**: os modelos de aprendizado de máquina dependem de hiperparâmetros bem ajustados para obter seu melhor desempenho. Assim, foram definidos os espaços de hiperparâmetros para três tipos de modelos: *RandomForestClassifier*, SVM e MLP. A busca por hiperparâmetros foi realizada por meio da técnica *RandomizedSearchCV*, que explora combinações aleatórias de hiperparâmetros dentro dos espaços definidos. Esse processo ocorre em um loop de validação cruzada de 5 divisões para cada modelo. Durante a busca, os modelos são ajustados aos dados de treinamento, e os hiperparâmetros são otimizados. Os resultados de desempenho foram apresentados em métricas, dentre elas: acurácia, MCC (*Matthews Correlation Coefficient*), Kappa, matriz de confusão e relatório de classificação.

- **seleção do melhor modelo)**: foi escolhido com base em uma métrica de pontuação definida. Essa estratégia permite reter o modelo que atinge a maior pontuação, garantindo a qualidade do modelo final. Vale ressaltar que esse processo é aplicado a cada tipo de modelo (*RandomForestClassifier*, SVM e MLP), resultando nos melhores modelos de cada categoria.

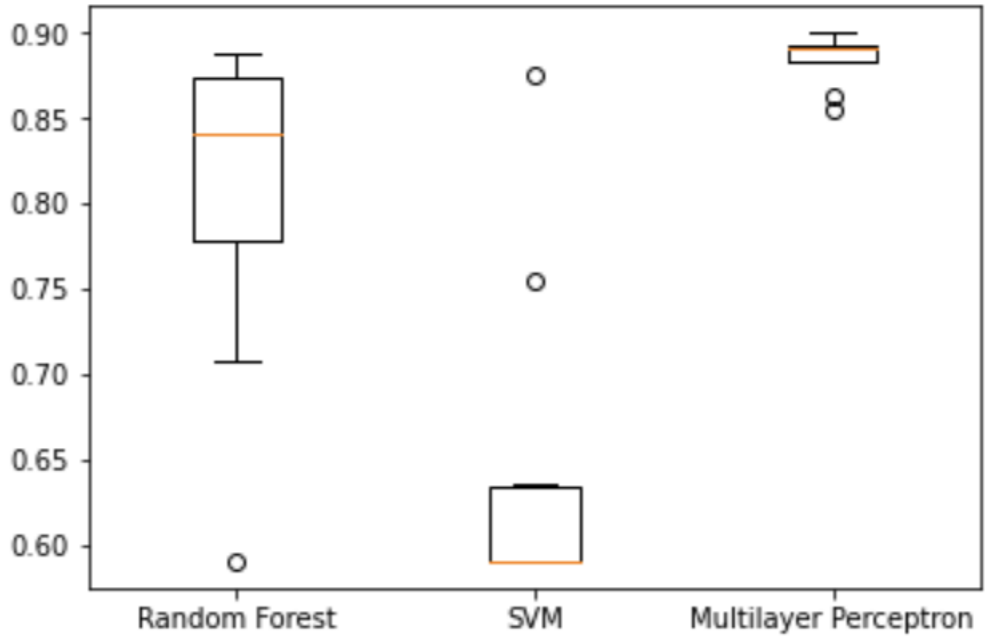
- **tuner para redes neurais (Keras Tuner)**: a busca por hiperparâmetros em redes neurais é uma tarefa desafiadora, para a qual foi utilizado o Keras Tuner. A função "*build_model*" foi configurada para criar os modelos de redes neurais com hiperparâmetros ajustáveis, como o número de unidades e a taxa de aprendizado. Um tuner é configurado com a técnica *RandomSearch*, que explora combinações promissoras de hiperparâmetros para a rede neural. A busca é limitada a um número específico de tentativas (5), com várias execuções por tentativa (3). Os melhores modelos de redes neurais identificados pelo tuner são armazenados como "*best_models*".

5.2.4 Validação dos modelos usando conjuntos de teste

5.2.4.1 Descritores Morgan: Dados de treinamentos e testes

A Figura 35 apresenta a comparação dos três modelos: MLP, SVM e *Random Forest* usando os descritores de Morgan.

Figura 35 – Comparação dos modelos. Fonte: Autoria própria.



Sobre o MLP, a melhor pontuação média foi 0.8996, o que indica um desempenho muito bom. As médias das pontuações em diferentes configurações de hiperparâmetros variaram, mas em geral, foram altas, acima de 0.85. Esse resultado demonstra que o MLP foi consistente em fornecer bom desempenho em várias configurações. A Tabela 5 apresenta o resultado da validação cruzada estratificada para o algoritmo MLP e descritor Morgan.

Em relação ao SVM, a melhor pontuação média foi 0.8747, o que também é uma pontuação considerável. No entanto, as médias das pontuações em algumas configurações foram baixas, por exemplo, 0.5899. Esse resultado indica que o desempenho do SVM pode ter sido muito sensível aos hiperparâmetros, e algumas configurações podem não ter funcionado bem. A Tabela 6 apresenta o resultado da validação cruzada estratificada para o algoritmo SVM e descritor Morgan.

Já o *Random Forest*, a melhor pontuação média foi 0.8882, que ficou entre as pontuações do MLP e do SVM. Assim como o SVM, o desempenho do *Random Forest* variou em diferentes configurações, com algumas tendo médias mais baixas. O *Random Forest* também mostrou sensibilidade aos hiperparâmetros, mas em geral, foi uma escolha sólida. A Tabela 7 apresenta o resultado da validação cruzada estratificada para o algoritmo *Random Forest* e descritor Morgan.

A busca pelos melhores hiperparâmetros se faz necessária para a obtenção de desempenho otimizado (WU *et al.*, 2019). A aplicação de descritores Morgan em cada um dos modelos de aprendizado de máquina (MLP, SVM e o *Random Forest*) revelou suas configurações ideais, demonstrando a importância do refinamento dos hiperparâmetros para extrair o máximo potencial desses modelos na análise de dados químicos e biológicos.

Tabela 5 – Score médio e desvio padrão para os hiperparâmetros treinados e testados na validação cruzada estratificada para o algoritmo MLP e descritor Morgan

Rank	Configuração	Score Médio	Desvio Padrão
1	activation: relu, alpha: 0.01, hidden_layer_sizes: 24, learning_rate: adaptive, max_iter: 2000, solver: adam	0.8996	0.0084
2	activation: tanh, alpha: 1.0, hidden_layer_sizes: 98, learning_rate: constant, max_iter: 2000, solver: sgd	0.8918	0.0088
3	activation: logistic, alpha: 0.1, hidden_layer_sizes: 92, learning_rate: adaptive, max_iter: 2000, solver: sgd	0.8931	0.0056
4	activation: tanh, alpha: 10.0, hidden_layer_sizes: 71, learning_rate: adaptive, max_iter: 2000, solver: adam	0.8892	0.0097
5	activation: logistic, alpha: 0.001, hidden_layer_sizes: 16, learning_rate: constant, max_iter: 2000, solver: sgd	0.8915	0.0080
6	activation: relu, alpha: 0.01, hidden_layer_sizes: 73, learning_rate: constant, max_iter: 2000, solver: sgd	0.8915	0.0080
7	activation: tanh, alpha: 100.0, hidden_layer_sizes: 69, learning_rate: constant, max_iter: 2000, solver: sgd	0.8853	0.0072
8	activation: relu, alpha: 10.0, hidden_layer_sizes: 51, learning_rate: adaptive, max_iter: 2000, solver: adam	0.8967	0.0090
9	activation: relu, alpha: 100.0, hidden_layer_sizes: 33, learning_rate: adaptive, max_iter: 2000, solver: adam	0.8545	0.0097
10	activation: logistic, alpha: 0.0001, hidden_layer_sizes: 97, learning_rate: invscaling, max_iter: 2000, solver: adam	0.8825	0.0101

Tabela 6 – Score médio e desvio padrão para os hiperparâmetros treinados e testados na validação cruzada estratificada para o algoritmo SVM e descritor Morgan

Rank	Configuração	Score Médio	Desvio Padrão
1	C: 9756.896309824398, gamma: 3.064599841241146e-05, kernel: rbf	0.8747	0.0125
2	C: 608.0332116863503, gamma: 0.015509913987594298, kernel: rbf	0.8765	0.0071
3	C: 0.15252471554120095, gamma: 0.00010929592787219392, kernel: rbf	0.7631	0.0135
4	C: 16.344819951627372, gamma: 0.09047071957568387, kernel: rbf	0.8405	0.0056
5	C: 7.4511565022821e-05, gamma: 1.235838277230692e-05, kernel: rbf	0.7072	0.0121
6	C: 0.004476173538513515, gamma: 0.004712973756110781, kernel: rbf	0.8395	0.0071
7	C: 4.977409198051348e-06, gamma: 1.1567327199145976, kernel: rbf	0.5899	0.0004
8	C: 0.00015201960735785719, gamma: 1.9223460470643646e-05, kernel: rbf	0.5899	0.0004
9	C: 0.015509913987594298, gamma: 6.156997328235204, kernel: rbf	0.6283	0.0046
10	C: 0.00010929592787219392, gamma: 0.09047071957568387, kernel: rbf	0.7546	0.0070

- MLP
  - Melhor pontuação (*best_score*): 0.8996
  - Melhores parâmetros (*best_params*):
    - função de ativação (*activation*): "*relu*"
    - *alpha*: 0.01
    - número de neurônios nas camadas ocultas (*hidden_layer_sizes*): 24
    - taxa de aprendizado (*learning_rate*): "*adaptive*"
    - número máximo de iterações (*max_iter*): 2000
    - *Solver*: "*adam*"
  - Pontuação Média nos Testes (*mean_test_score*): [0.8996, 0.8931, 0.8545, 0.8825, 0.8618, 0.8853, 0.8967, 0.8892, 0.8918, 0.8915]
- SVM
  - Melhores parâmetros (*best_params*):
    - parâmetro C: 9756.8963
    - parâmetro gamma: 3.0646e-05
    - *kernel*: "rbf"

Tabela 7 – Score médio e desvio padrão para os hiperparâmetros treinados e testados na validação cruzada estratificada para o algoritmo *Random Forest* e descritor Morgan

Rank	Configuração	Score Médio	Desvio Padrão
1	bootstrap: False, criterion: entropy, max_depth: 18, max_features: 292, min_samples_leaf: 9, min_samples_split: 3, n_estimators: 439	0.8882	0.0052
2	bootstrap: True, criterion: entropy, max_depth: 15, max_features: 409, min_samples_leaf: 8, min_samples_split: 8, n_estimators: 221	0.8744	0.0072
3	bootstrap: True, criterion: gini, max_depth: 11, max_features: 409, min_samples_leaf: 4, min_samples_split: 9, n_estimators: 763	0.8765	0.0071
4	bootstrap: False, criterion: entropy, max_depth: 1, max_features: 682, min_samples_leaf: 12, min_samples_split: 18, n_estimators: 574	0.5899	0.0004
5	bootstrap: True, criterion: gini, max_depth: 10, max_features: 682, min_samples_leaf: 16, min_samples_split: 16, n_estimators: 289	0.8405	0.0056
6	bootstrap: False, criterion: gini, max_depth: 19, max_features: 682, min_samples_leaf: 20, min_samples_split: 4, n_estimators: 584	0.8729	0.0047
7	bootstrap: True, criterion: gini, max_depth: 9, max_features: 409, min_samples_leaf: 18, min_samples_split: 5, n_estimators: 700	0.8237	0.0063
8	bootstrap: False, criterion: gini, max_depth: 7, max_features: 682, min_samples_leaf: 8, min_samples_split: 16, n_estimators: 134	0.8395	0.0070
9	bootstrap: True, criterion: entropy, max_depth: 2, max_features: 682, min_samples_leaf: 12, min_samples_split: 7, n_estimators: 485	0.7072	0.0121
10	bootstrap: True, criterion: gini, max_depth: 4, max_features: 292, min_samples_leaf: 8, min_samples_split: 5, n_estimators: 101	0.7631	0.0135

  - melhor pontuação (*best_score*): 0.8747
  - pontuação média nos testes (*mean_test_score*): [0.58995603, 0.6355177, 0.58995603, 0.58995603, 0.62826951, 0.58995603, 0.87471596, 0.58995603, 0.58995603, 0.75459518]

- *Random Forest*
  - Melhores parâmetros (*best_params*):
    - *bootstrap*: *false*
    - critério de divisão (*criterion*): "*entropy*"
    - profundidade máxima da árvore (*max_depth*): 18
    - máximo de *features* (*max_features*): 292
    - mínimo de amostras em folhas (*min_samples_leaf*): 9
    - mínimo de amostras em nós internos (*min_samples_split*): 3
    - número de estimadores (*n_estimators*): 439
  - Melhor pontuação (*best_score*): 0.8882
  - Pontuação Média nos Testes (*mean_test_score*): [0.87444986, 0.87652106, 0.70722305, 0.58995603, 0.8405365, 0.87289613, 0.82371053, 0.88817004, 0.83950559, 0.76313837]

Em relação ao *Tensorflow*, temos os seguintes resultados:

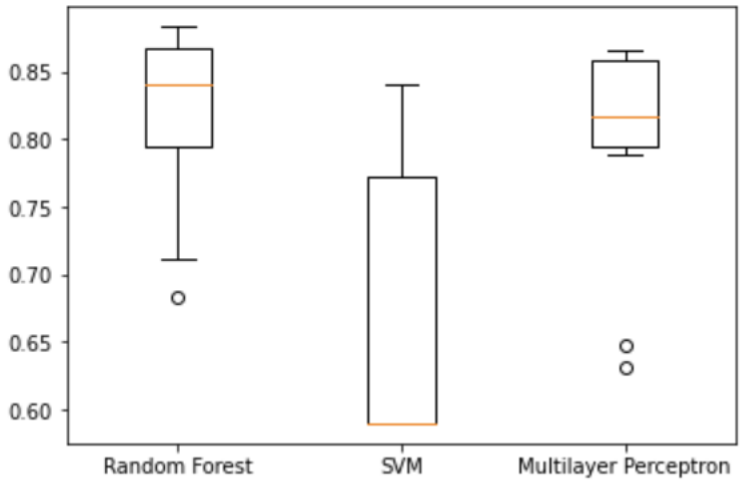
- *trial 5 complete*: após 13 segundos de execução, a quinta tentativa (*Trial* 5) da pesquisa de hiperparâmetros foi concluída.
- precisão de validação (*val_accuracy*): a precisão de validação da *Trial* 5 foi de aproximadamente 0.8870, indicando o desempenho do modelo nesta tentativa específica.
- melhor precisão de validação até o momento: foi de cerca de 0.9107, o que sugere que a *Trial* 5 não superou o melhor desempenho anterior.
- tempo total decorrido: o tempo total decorrido durante todo o processo de pesquisa de hiperparâmetros foi de 53 segundos.
- hiperparâmetros otimizados: os melhores hiperparâmetros encontrados são representados por um objeto *hyperParameters*. estes hiperparâmetros específicos não foram fornecidos na saída.
- melhor modelo encontrado: o melhor modelo identificado é uma rede neural sequencial com a seguinte arquitetura:
  - camada densa 1 com 416 neurônios.

- camada densa 2 com 96 neurônios.
- camada densa 3 com 1 neurônio.
- total de parâmetros no modelo: 892,513.
- todos os parâmetros são treináveis (*Trainable params*: 892,513), o que significa que o modelo pode ser ajustado durante o treinamento.
- não há parâmetros não treináveis (*Non-trainable params*: 0) no modelo.

5.2.4.2 Descritores SiRMS: Dados de treinamentos e teste

A Figura 36 apresenta a comparação dos três modelos: MLP, SVM e Random Forest usando os descritores SiRMS.

Figura 36 – Comparação dos modelos. Fonte: Autoria própria.



O modelo MLP obteve uma média de pontuações durante a busca por hiperparâmetros, variando entre 0.6306 e 0.8693. Essa variação evidencia a sensibilidade do desempenho do modelo MLP às diferentes configurações de hiperparâmetros testadas. A melhor pontuação encontrada foi de aproximadamente 0.8693, destacando o potencial do modelo em sua melhor configuração. A Tabela 8 apresenta o resultado da validação cruzada estratificada para o algoritmo MLP e descritor SiRMS.

Já no modelo SVM, observamos que as médias de pontuações variaram entre 0.5899 e 0.8400 durante a busca por hiperparâmetros. Essa variação sugere que o desempenho do modelo SVM também variou consideravelmente ao testar as diversas combinações de hiperparâmetros. A melhor pontuação encontrada foi de cerca de 0.8400. A Tabela 9 apresenta o resultado da validação cruzada estratificada para o algoritmo SVM e descritor SiRMS.

Durante a exploração dos hiperparâmetros, o modelo *Random Forest* apresentou médias de pontuações, variando de 0.6024 a 0.8364. Assim como nos outros modelos, as

Tabela 8 – Score médio e desvio padrão para os hiperparâmetros treinados e testados na validação cruzada estratificada para o algoritmo MLP e descritor SiRMS

Rank	Configuração	Score Médio	Desvio Padrão
1	activation: logistic, alpha: 0.0001, hidden_layer_sizes: 97, learning_rate: invscaling, max_iter: 2000, solver: adam	0.865907	0.0116504
2	activation: relu, alpha: 0.01, hidden_layer_sizes: 24, learning_rate: adaptive, max_iter: 2000, solver: adam	0.859437	0.018464
3	activation: relu, alpha: 0.01, hidden_layer_sizes: 73, learning_rate: constant, max_iter: 2000, solver: sgd	0.858662	0.00965879
4	activation: tanh, alpha: 1.0, hidden_layer_sizes: 98, learning_rate: constant, max_iter: 2000, solver: sgd	0.85581	0.0148686
5	activation: logistic, alpha: 0.001, hidden_layer_sizes: 16, learning_rate: constant, max_iter: 2000, solver: sgd	0.820092	0.0193891
6	activation: relu, alpha: 10.0, hidden_layer_sizes: 51, learning_rate: adaptive, max_iter: 2000, solver: adam	0.812843	0.0245042
7	activation: relu, alpha: 100.0, hidden_layer_sizes: 33, learning_rate: adaptive, max_iter: 2000, solver: adam	0.631635	0.0103565
8	activation: logistic, alpha: 0.1, hidden_layer_sizes: 92, learning_rate: adaptive, max_iter: 2000, solver: sgd	0.812584	0.0168114
9	activation: tanh, alpha: 100.0, hidden_layer_sizes: 69, learning_rate: constant, max_iter: 2000, solver: sgd	0.647167	0.00662246
10	activation: tanh, alpha: 1.0, hidden_layer_sizes: 98, learning_rate: constant, max_iter: 2000, solver: sgd	0.820092	0.0193891

Tabela 9 – Score médio e desvio padrão para os hiperparâmetros treinados e testados na validação cruzada estratificada para o algoritmo SVM e descritor SiRMS

Rank	Configuração	Score Médio	Desvio Padrão
1	C: 0.0155099, gamma: 0.0155099, kernel: rbf	0.84002	0.00611567
2	C: 0.00447617, gamma: 0.00471297, kernel: rbf	0.824746	0.00859098
3	C: 608.033, gamma: 0.0155099, kernel: rbf	0.869012	0.0149086
4	C: 0.152525, gamma: 0.000109296, kernel: rbf	0.781768	0.0318786
5	C: 16.3448, gamma: 0.0904707, kernel: rbf	0.844679	0.0161889
6	C: 1.76609, gamma: 6.157, kernel: rbf	0.863319	0.0137963
7	C: 0.0312235, gamma: 4.51856, kernel: rbf	0.874706	0.0129533
8	C: 0.00015202, gamma: 1.92235e-05, kernel: rbf	0.883253	0.0103789
9	C: 7.45116e-05, gamma: 1.23584e-05, kernel: rbf	0.711621	0.0161448
10	C: 0.00015202, gamma: 0.000109296, kernel: rbf	0.615845	0.0056231

pontuações mostraram variações significativas à medida que diferentes configurações de hiperparâmetros foram avaliadas. A melhor pontuação encontrada para o modelo *Random Forest* foi de aproximadamente 0.8364. A Tabela 10 apresenta o resultado da validação cruzada estratificada para o algoritmo *Random Forest* e descritor SiRMS.

Os melhores hiperparâmetros utilizando os descritores SiRMS foram:

- *Random Forest*
  - Melhor pontuação (*best_score*): 0.8364
  - Melhores parâmetros (*best_params*): "*bootstrap*": *True*, "*criterion*": "*gini*", "*max_depth*": 18, "*max_features*": "*auto*", "*min_samples_leaf*": 14, "*min_samples_split*": 19, "*n_estimators*": 200
  - Pontuações médias nos testes (*mean_test_score*): [0.7916, 0.7958, 0.6539, 0.8348, 0.7432, 0.6997, 0.8364, 0.6024, 0.7051, 0.6529]
- MLP
  - Melhor pontuação (*best_score*): 0.8693
  - Melhores parâmetros (*best_params*): "*activation*": "*logistic*", "*alpha*": 0.0001, "*hidden_layer_sizes*": 97, "*learning_rate*": "*invscaling*", "*max_iter*": 2000, "*solver*": "*adam*"

Tabela 10 – Score médio e desvio padrão para os hiperparâmetros treinados e testados na validação cruzada estratificada para o algoritmo *Random Forest* e descritor SiRMS

Rank	Configuração	Score Médio	Desvio Padrão
1	criterion: entropy, max_depth: 18, max_features: 197, min_samples_leaf: 9, min_samples_split: 3, n_estimators: 439	0.883253	0.0103789
2	criterion: entropy, max_depth: 15, max_features: 276, min_samples_leaf: 8, min_samples_split: 8, n_estimators: 221	0.874706	0.0129533
3	criterion: gini, max_depth: 11, max_features: 276, min_samples_leaf: 4, min_samples_split: 9, n_estimators: 763	0.869012	0.0149086
4	criterion: entropy, max_depth: 1, max_features: 461, min_samples_leaf: 12, min_samples_split: 18, n_estimators: 574	0.683922	0.0156668
5	criterion: gini, max_depth: 10, max_features: 461, min_samples_leaf: 16, min_samples_split: 16, n_estimators: 289	0.844679	0.0161889
6	criterion: gini, max_depth: 9, max_features: 276, min_samples_leaf: 18, min_samples_split: 5, n_estimators: 700	0.834324	0.0135906
7	criterion: gini, max_depth: 7, max_features: 461, min_samples_leaf: 8, min_samples_split: 16, n_estimators: 134	0.835878	0.0179037
8	criterion: gini, max_depth: 4, max_features: 197, min_samples_leaf: 8, min_samples_split: 5, n_estimators: 101	0.781768	0.0318786
9	criterion: entropy, max_depth: 2, max_features: 461, min_samples_leaf: 12, min_samples_split: 7, n_estimators: 485	0.711621	0.0161448
10	criterion: entropy, max_depth: 1, max_features: 461, min_samples_leaf: 12, min_samples_split: 18, n_estimators: 574	0.683922	0.0156668

  - Pontuações médias nos testes (*mean_test_score*): [0.8633, 0.8113, 0.6306, 0.8693, 0.6464, 0.8475, 0.8066, 0.7893, 0.8594, 0.8183]

- SVM

  - Melhores parâmetros (*best_params*): "C": 608.03, "*gamma*": 0.0155, "*kernel*": "*rbf*"
  - Melhor pontuação (*best_score*): 0.8400
  - Pontuações médias nos testes (*mean_test_score*): [0.5900, 0.8400, 0.5900, 0.5900, 0.8247, 0.5900, 0.8341, 0.5900, 0.5900, 0.6158]

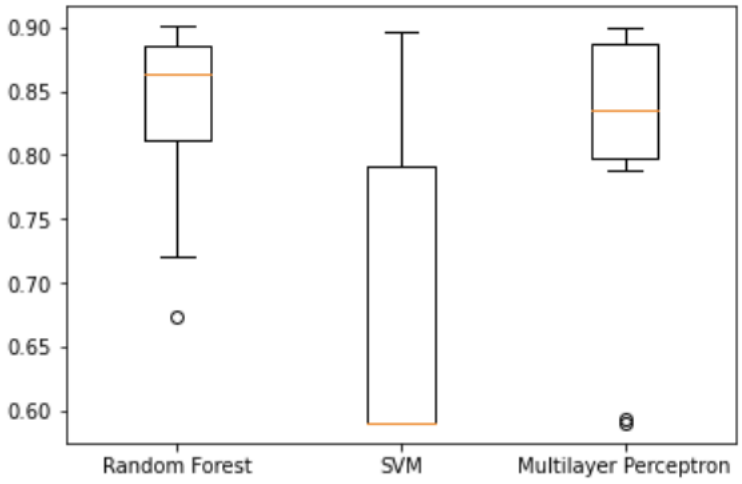
Os seguintes resultados foram obtidos em relação ao *Tensorflow*:

- *trial 5 complete*: após 10 segundos de execução, a quinta tentativa (*Trial* 5) da pesquisa de hiperparâmetros foi concluída.
- precisão de validação (*val_accuracy*): a precisão de validação da *Trial* 5 foi de aproximadamente 0.8064, o que indica o desempenho do modelo nessa tentativa específica.
- melhor precisão de validação até o momento: foi de cerca de 0.8461, o que indica que a *Trial* 5 não superou o melhor desempenho anterior.
- tempo total decorrido: o tempo total decorrido durante todo o processo de pesquisa de hiperparâmetros foi de 39 segundos.
- hiperparâmetros otimizados: os melhores hiperparâmetros encontrados são representados por um objeto *hyperParameters*. estes hiperparâmetros específicos não foram fornecidos na saída.
- melhor modelo encontrado: o melhor modelo identificado foi uma rede neural sequencial com a seguinte arquitetura:
  - camada densa 1 com 192 neurônios.
  - camada densa 2 com 32 neurônios.
  - camada densa 3 com 1 neurônio.
  - total de parâmetros no modelo: 272,129.
  - todos os parâmetros são treináveis (*trainable params*: 272,129), indicando que o modelo pode ser ajustado durante o treinamento.
  - não há parâmetros não treináveis (*Non-trainable params*: 0) no modelo.

5.2.4.3 Descritores *RDKit*: Dados de treinamentos e teste

A Figura 37 apresenta a comparação dos três modelos: MLP, SVM e *Random Forest* usando os descritores *RDKit*.

Figura 37 – Comparação dos modelos. Fonte: Autoria própria.



Durante as buscas por hiperparâmetros, o modelo MLP apresentou uma média de pontuação que variou de 0.5899 a 0.8996. Essa grande variação indica que o desempenho do modelo MLP foi significativamente influenciado pelas diferentes configurações de hiperparâmetros testadas. A melhor pontuação obtida foi aproximadamente 0.8996. A Tabela 11 apresenta o resultado da validação cruzada estratificada para o algoritmo MLP e descritor RDKit.

No caso do modelo SVM, as médias das pontuações variaram de 0.5899 a 0.8996 durante a busca por hiperparâmetros. Isso indica que o desempenho do modelo SVM também foi sensível às diferentes combinações de hiperparâmetros testadas. A melhor pontuação encontrada foi de cerca de 0.8400. A Tabela 12 apresenta o resultado da validação cruzada estratificada para o algoritmo SVM e descritor RDKit.

O modelo *Random Forest* apresentou médias de pontuações variando de 0.5899 a 0.8996 durante a busca por hiperparâmetros. Da mesma forma que nos outros modelos, as pontuações variaram à medida que diferentes configurações de hiperparâmetros foram avaliadas. A melhor pontuação alcançada para o modelo *Random Forest* foi de, aproximadamente, 0.8364. A Tabela 13 apresenta o resultado da validação cruzada estratificada para o algoritmo *Random Forest* e descritor RDKit.

Os melhores hiperparâmetros utilizando os descritores *RDKit* foram:

- *Random Forest*
  - Melhor pontuação (*best_score*): 0.9008

Tabela 11 – Score médio e desvio padrão para os hiperparâmetros treinados e testados na validação cruzada estratificada para o algoritmo MLP e descritor RDKit

Rank	Configuração	Score Médio	Desvio Padrão
1	activation: logistic, alpha: 0.1, hidden_layer_sizes: 92, learning_rate: adaptive, max_iter: 2000, solver: sgd	0.82913821	0.02317217
2	activation: logistic, alpha: 0.001, hidden_layer_sizes: 16, learning_rate: constant, max_iter: 2000, solver: sgd	0.84183318	0.01533405
3	activation: gini, alpha: 100.0, hidden_layer_sizes: 98, learning_rate: constant, max_iter: 2000, solver: sgd	0.88739418	0.01399989
4	activation: tanh, alpha: 1.0, hidden_layer_sizes: 98, learning_rate: constant, max_iter: 2000, solver: sgd	0.88894657	0.01279425
5	activation: relu, alpha: 0.01, hidden_layer_sizes: 24, learning_rate: adaptive, max_iter: 2000, solver: adam	0.88506157	0.00596097
6	activation: relu, alpha: 10.0, hidden_layer_sizes: 51, learning_rate: adaptive, max_iter: 2000, solver: adam	0.82914927	0.00920324
7	activation: tanh, alpha: 10.0, hidden_layer_sizes: 71, learning_rate: adaptive, max_iter: 2000, solver: adam	0.78747126	0.00999211
8	activation: tanh, alpha: 100.0, hidden_layer_sizes: 69, learning_rate: constant, max_iter: 2000, solver: sgd	0.59254402	0.00232471
9	activation: logistic, alpha: 0.01, hidden_layer_sizes: 24, learning_rate: adaptive, max_iter: 2000, solver: adam	0.58995603	0.00041358
10	activation: logistic, alpha: 0.0001, hidden_layer_sizes: 97, learning_rate: invscaling, max_iter: 2000, solver: adam	0.58990944	0.00041358

Tabela 12 – Score médio e desvio padrão para os hiperparâmetros treinados e testados na validação cruzada estratificada para o algoritmo SVM e descritor RDKit

Rank	Configuração	Score Médio	Desvio Padrão
1	C: 608.0332116863503, gamma: 0.015509913987594298, kernel: rbf	0.89645182	0.00987062
2	C: 0.03122348867288777, gamma: 4.518560951024106, kernel: rbf	0.8881687	0.01412344
3	C: 16.344819951627372, gamma: 0.09047071957568387, kernel: rbf	0.87134239	0.0079408
4	C: 0.15252471554120095, gamma: 0.00010929592787219392, kernel: rbf	0.79574902	0.02131956
5	C: 4.977409198051348e-06, gamma: 1.1567327199145976, kernel: rbf	0.67330333	0.01679328
6	C: 0.00015201960735785719, gamma: 1.9223460470643646e-05, kernel: rbf	0.58995603	0.00041358
7	C: 7.4511565022821e-05, gamma: 1.235838277230692e-05, kernel: rbf	0.58995603	0.00041358
8	C: 1.7660944735776943e-06, gamma: 6.156997328235204, kernel: rbf	0.58995603	0.00041358
9	C: 9756.896309824398, gamma: 3.064599841241146e-05, kernel: rbf	0.85865781	0.01621948
10	C: 0.004476173538513515, gamma: 0.004712973756110781, kernel: rbf	0.85788128	0.01694509

  - Melhores parâmetros (*best_params*): "*bootstrap*": *False*, "*criterion*": "*entropy*", "*max_depth*": 18, "*max_features*": 29, "*min_samples_leaf*": 9, "*min_samples_split*": 3, "*n_estimators*": 439
  - Pontuações médias nos testes (*mean_test_score*): [0.8881687, 0.88791064, 0.71964186, 0.67330333, 0.86461267, 0.87703752, 0.85788128, 0.90085395, 0.86383279, 0.79574902]

- MLP
  - Melhor pontuação (*best_score*): 0.8995
  - Melhores parâmetros (*best_params*): "*activation*": "*logistic*", "*alpha*": 0.0001, "*hidden_layer_sizes*": 97, "*learning_rate*": "*invscaling*", "*max_iter*": 2000, "*solver*": "*adam*"
  - Pontuações médias nos testes (*mean_test_score*): [0.88506157, 0.82913821, 0.58995603, 0.89955794, 0.59254402, 0.88739418, 0.82914927, 0.78747126, 0.88894657, 0.84183318]

- SVM
  - Melhores parâmetros (*best_params*): "C": 608.0332116863503, "*gamma*": 0.015509913987594298, "*kernel*": "*rbf*"

Tabela 13 – Score médio e desvio padrão para os hiperparâmetros treinados e testados na validação cruzada estratificada para o algoritmo *Random Forest* e descritor RDKit

Rank	Configuração	Score Médio	Desvio Padrão
1	bootstrap: False, criterion: entropy, max_depth: 18, max_features: 29, min_samples_leaf: 9, min_samples_split: 3, n_estimators: 439	0.90085395	0.01282782
2	bootstrap: False, criterion: entropy, max_depth: 1, max_features: 69, min_samples_leaf: 12, min_samples_split: 18, n_estimators: 574	0.86461267	0.01323663
3	bootstrap: True, criterion: entropy, max_depth: 15, max_features: 41, min_samples_leaf: 8, min_samples_split: 8, n_estimators: 221	0.8881687	0.01412344
4	bootstrap: True, criterion: gini, max_depth: 11, max_features: 41, min_samples_leaf: 4, min_samples_split: 9, n_estimators: 763	0.88791064	0.01460724
5	bootstrap: True, criterion: gini, max_depth: 10, max_features: 69, min_samples_leaf: 16, min_samples_split: 16, n_estimators: 289	0.87703752	0.01434483
6	bootstrap: False, criterion: gini, max_depth: 4, max_features: 29, min_samples_leaf: 8, min_samples_split: 5, n_estimators: 101	0.79574902	0.02131956
7	bootstrap: True, criterion: gini, max_depth: 9, max_features: 41, min_samples_leaf: 18, min_samples_split: 5, n_estimators: 700	0.85788128	0.01694509
8	bootstrap: True, criterion: entropy, max_depth: 2, max_features: 69, min_samples_leaf: 12, min_samples_split: 7, n_estimators: 485	0.71964186	0.01739644
9	bootstrap: True, criterion: gini, max_depth: 9, max_features: 41, min_samples_leaf: 8, min_samples_split: 16, n_estimators: 134	0.86383279	0.01694509
10	bootstrap: True, criterion: gini, max_depth: 2, max_features: 69, min_samples_leaf: 12, min_samples_split: 7, n_estimators: 485	0.67330333	0.01679328

  - Melhor pontuação (*best_score*): 0.8964
  - Pontuações médias nos testes (*mean_test_score*): [0.58995603, 0.89645182, 0.58995603, 0.58995603, 0.87134239, 0.58995603, 0.85865781, 0.58995603, 0.58995603, 0.59021476]

Os seguintes resultados foram obtidos em relação ao *Tensorflow*:

- *trial 5 complete*: após 13 segundos de execução, a quinta tentativa (*Trial* 5) da pesquisa de hiperparâmetros foi concluída.
- precisão de validação (*val_accuracy*): a precisão de validação da *Trial* 5 foi de aproximadamente 0.8870, indicando um desempenho promissor do modelo nesta tentativa específica.
- melhor precisão de validação até o momento: a melhor precisão de validação encontrada até o momento da pesquisa foi de cerca de 0.9107. Portanto, a *Trial* 5 não conseguiu superar o melhor desempenho anterior.
- tempo total decorrido: o tempo total decorrido durante todo o processo de pesquisa de hiperparâmetros foi de 53 segundos, mostrando eficiência na otimização dos hiperparâmetros.
- hiperparâmetros otimizados: os melhores hiperparâmetros encontrados são representados por um objeto *HyperParameters* cujos detalhes específicos não foram fornecidos na saída.
- melhor modelo encontrado: o melhor modelo identificado foi uma rede neural sequencial com a seguinte arquitetura:
  - camada densa 1 com 416 neurônios.
  - camada densa 2 com 96 neurônios.
  - camada densa 3 com 1 neurônio.
  - total de parâmetros no modelo: 892,513.
  - o total de parâmetros no modelo é de 892,513, todos eles treináveis, o que significa que o modelo pode ser ajustado durante o treinamento. Não haviam parâmetros não treináveis no modelo. Este modelo parece ser bastante complexo e pode ter a capacidade de capturar padrões complexos nos dados.

5.3 Seleção e validação dos modelos

Esta etapa visa realizar a avaliação externa de modelos de classificação QSAR e considerar a questão do domínio de aplicabilidade (AD) desses modelos. A avaliação

externa foi realizada em um conjunto de validação externa, que compreendeu 20% dos dados do conjunto de dados total.

Para tanto, foi realizado o cálculo de várias métricas de desempenho do modelo de classificação com base nas previsões (*y_pred*) e nos rótulos verdadeiros (*y_test*) do conjunto de validação externa. As métricas incluem: o coeficiente Kappa, Área Sob a Curva (AUC), sensibilidade, precisão, especificidade, valor preditivo negativo, acurácia, *F1 Score* e cobertura. Essas métricas foram utilizadas para avaliar o desempenho geral do modelo em relação aos dados de validação externa.

A obtenção dessas métricas envolveu a realização de uma validação cruzada estratificada de 5 *folds* (BEY *et al.*, 2020), com a mesma configuração utilizada para treinamento e testes, no conjunto de validação externa para avaliar o modelo de classificação (m). O procedimento envolveu as seguintes etapas:

- divisão dos dados de validação externa estratificada em 5 *folds*.
- treinamento do modelo (m) em 4 dos 5 *folds*, com avaliação no *fold* restante, repetindo o processo cinco vezes (uma para cada *fold*).
- coleta das previsões (*fold_pred*) e dos valores de domínio de aplicabilidade (*fold_ad*) para cada *fold*.
- aplicação de um limite (*threshold_ad*) aos valores de AD para determinar se um exemplo está dentro do domínio de aplicabilidade ou não.
- combinação das previsões do modelo e dos valores de AD com base no limite, levando em consideração o domínio de aplicabilidade nas previsões.
- cálculo da cobertura (*coverage_5f*) para avaliar a proporção de exemplos que estão dentro do domínio de aplicabilidade.
- cálculo das métricas de desempenho do modelo (usando o Método 1) e das métricas de desempenho do modelo com AD em relação aos rótulos verdadeiros.
- apresentação das estatísticas de avaliação do modelo, incluindo métricas de desempenho e cobertura.

As tabelas e gráficos a seguir apresentam os valores correspondentes a cada métrica e modelo. Estes incluem os descritores Morgan (Tabela 14 e Figura 38), SiRMS (Tabela 15 e Figura 39) e RDKit (Tabela 16 e Figura 40), os quais foram aplicados aos algoritmos *Random Forest*, SVM, MLP e *TensorFlow*.

Tabela 14 – Métricas calculadas para cada modelo obtido utilizando o *Random forest*, SVM, MLP e TensorFlow com o descritor Morgan e o método de validação cruzada estratificada 5-*fold* para o conjunto de substâncias com atividade frente à enzima AChE

	Kappa	AUC	Sens.	PPV	Espec.	NPV	Acur.	*F-score*	Cober.
Random forest									
Morgan	0,61	0,81	0,81	0,85	0,81	0,75	0,81	0,83	1,00
Morgan AD	0,82	0,88	0,99	0,95	0,78	0,93	0,94	0,97	0,35
SVM									
Morgan	0,59	0,79	0,84	0,82	0,74	0,77	0,80	0,83	1,00
Morgan AD	0,76	0,88	0,93	0,91	0,83	0,86	0,89	0,92	0,63
MLP									
Morgan	0,65	0,82	0,87	0,84	0,78	0,81	0,83	0,86	1,00
Morgan AD	0,73	0,86	0,91	0,87	0,81	0,87	0,87	0,89	0,84
Tensorflow									
Morgan	0,73	0,86	0,89	0,88	0,84	0,84	0,87	0,89	

Figura 38 – Comparação de métricas para diferentes algoritmos e conjuntos de dados usando o descritor Morgan. Fonte: Autoria própria.

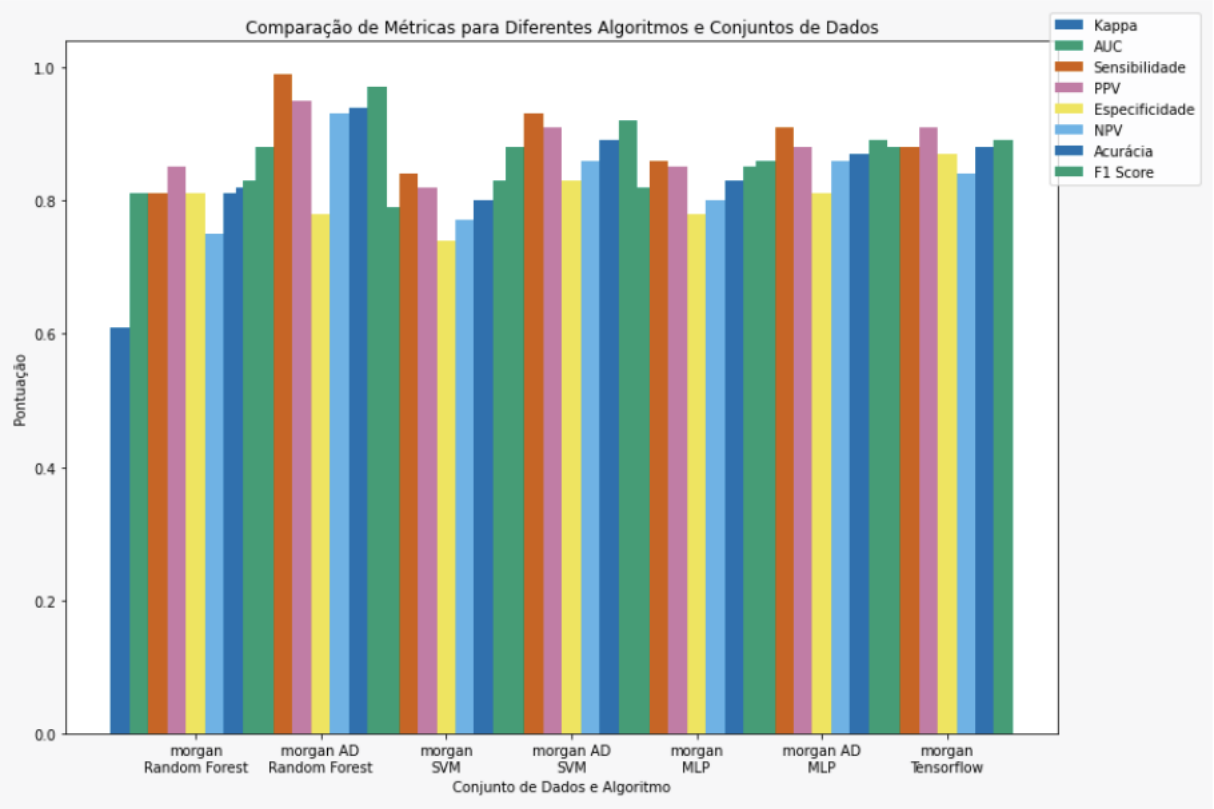


Tabela 15 – Métricas calculadas para cada modelo obtido utilizando o *Random forest*, SVM, MLP e TensorFlow com o descritor SiRMS e o método de validação cruzada estratificada 5-*fold* para o conjunto de substâncias com atividade frente à enzima AChE

	Kappa	AUC	Sens.	PPV	Espec.	NPV	Acur.	*F-score*	Cober.
Random forest									
Morgan	0,60	0,80	0,85	0,82	0,74	0,78	0,80	0,83	1,00
Morgan AD	0,88	0,93	0,98	0,95	0,88	0,95	0,95	0,96	0,41
SVM									
Morgan	0,53	0,76	0,84	0,78	0,68	0,76	0,77	0,81	1,00
Morgan AD	0,59	0,79	0,90	0,79	0,68	0,83	0,80	0,84	0,24
MLP									
Morgan	0,60	0,80	0,84	0,83	0,76	0,77	0,80	0,83	1,00
Morgan AD	0,68	0,84	0,89	0,85	0,79	0,84	0,85	0,87	0,82
Tensorflow									
Morgan	0,69	0,85	0,84	0,89	0,85	0,79	0,85	0,86	

Figura 39 – Comparação de métricas para diferentes algoritmos e conjuntos de dados usando o descritor SiRMS. Fonte: Autoria própria.

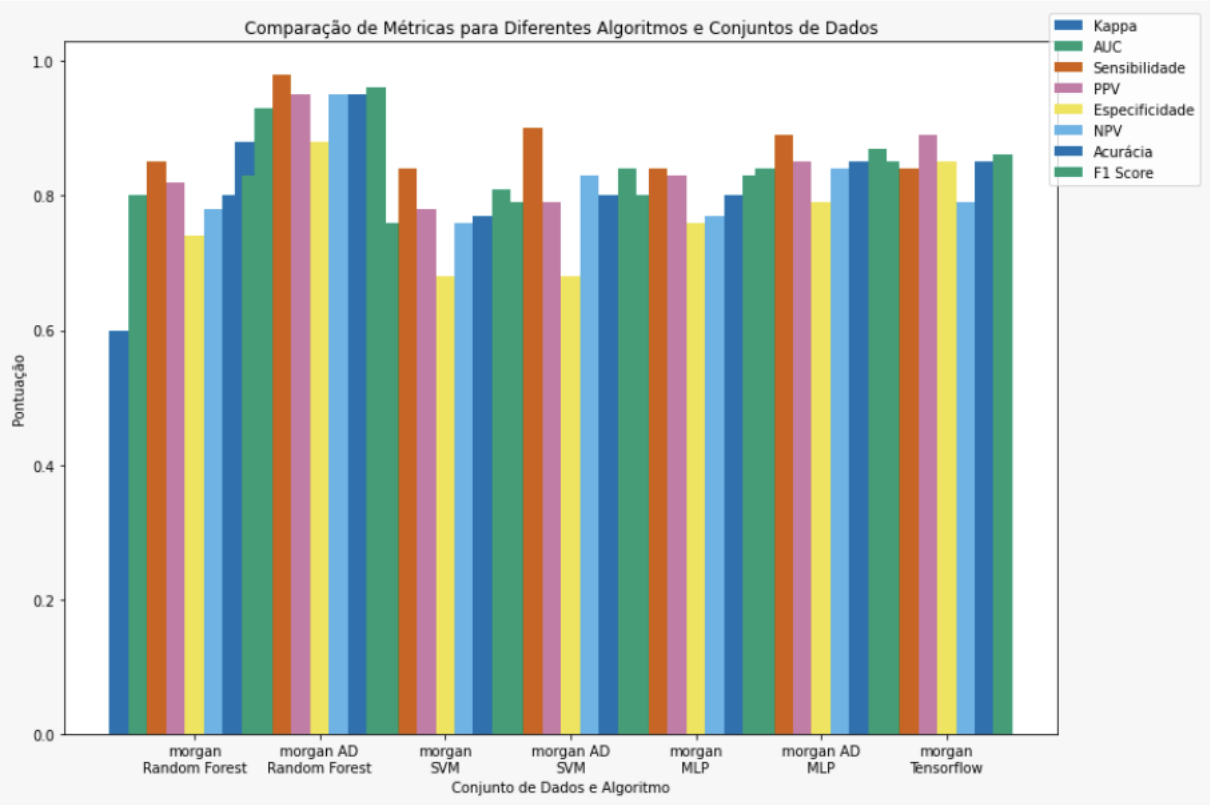
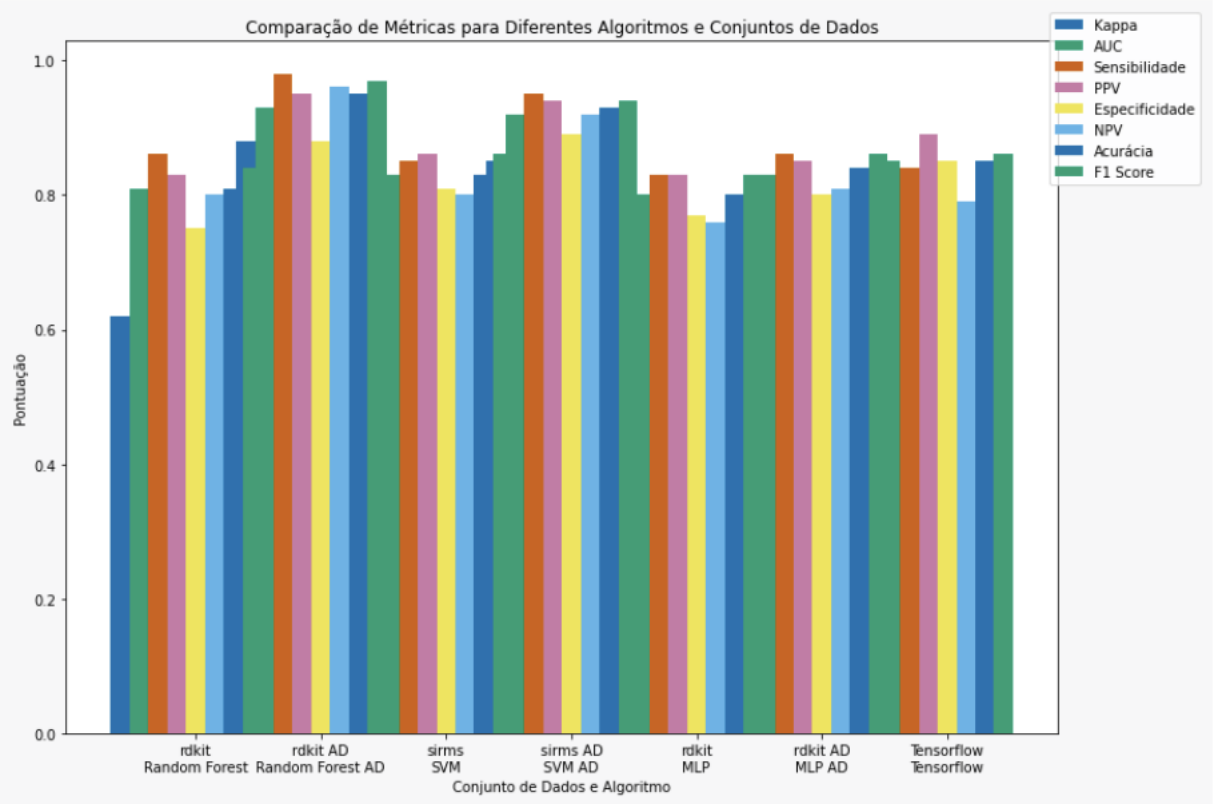


Tabela 16 – Métricas calculadas para cada modelo obtido utilizando o *Random forest*, SVM, MLP e TensorFlow com o descritor *RDKit* e o método de validação cruzada estratificada 5-*fold* para o conjunto de substâncias com atividade frente à enzima AChE. Em negrito estão destacados os melhores resultados.

	Kappa	AUC	Sens.	PPV	Espec.	NPV	Acur.	*F-score*	Cober.
Random forest									
Morgan	0,65	0,82	0,87	0,84	0,77	0,82	0,83	0,86	1,00
Morgan AD	**0,87**	**0,92**	**0,98**	**0,95**	**0,86**	**0,95**	**0,95**	**0,97**	0,35
SVM									
Morgan	0,62	0,81	0,84	0,84	0,78	0,78	0,81	0,84	1,00
Morgan AD	0,80	0,90	0,93	0,92	0,86	0,88	0,91	0,93	0,61
MLP									
Morgan	0,60	0,80	0,86	0,82	0,74	0,79	0,81	0,84	1,00
Morgan AD	0,71	0,85	0,90	0,87	0,80	0,85	0,86	0,89	**0,80**
Tensorflow									
Morgan	0,69	0,84	0,89	0,86	0,8	0,84	0,85	0,87	

Figura 40 – Comparação de métricas para diferentes algoritmos e conjuntos de dados usando o descritor RDKit. Fonte: Autoria própria.



Assim, esta etapa realizou uma avaliação externa dos modelos de classificação QSAR, usando um conjunto de validação externa e considerando o domínio de aplicabilidade por meio dos valores de AD. As métricas de desempenho foram calculadas tanto para

o modelo-base quanto para o modelo com AD, permitindo uma avaliação completa do desempenho do modelo em tarefas de classificação.

O teste de permutação foi realizado visando avaliar a significância estatística do desempenho dos modelos de classificação em relação ao conjunto de dados aleatório. Logo, os modelos de classificação treinados foram avaliados em dois cenários diferentes, sendo eles:

- **cenário 1: Dados reais**: a métrica de avaliação utilizada foi a acurácia, responsável por medir a precisão das previsões do modelo com base nos dados reais. O desempenho do modelo foi avaliado usando o escore (pontuações) real(is).

- **cenário 2: Dados aleatórios (*Random*)**: um conjunto de dados aleatório foi criado com a mesma quantidade de amostras e características, porém sem correlação entre eles. Esse conjunto de dados aleatórios também foi dividido em dois conjuntos usando a validação cruzada estratificada (*Stratified K-Fold*) com 2 dobras. O desempenho dos modelos foi, assim, avaliado usando as pontuações geradas a partir desse conjunto aleatório (*score_rand*).

Assim, para determinar se o desempenho do modelo com dados reais era estatisticamente significativo ao comparar com o desempenho obtido com dados aleatórios, foi realizado um **teste de permutação** (OJALA; GARRIGA, 2010).

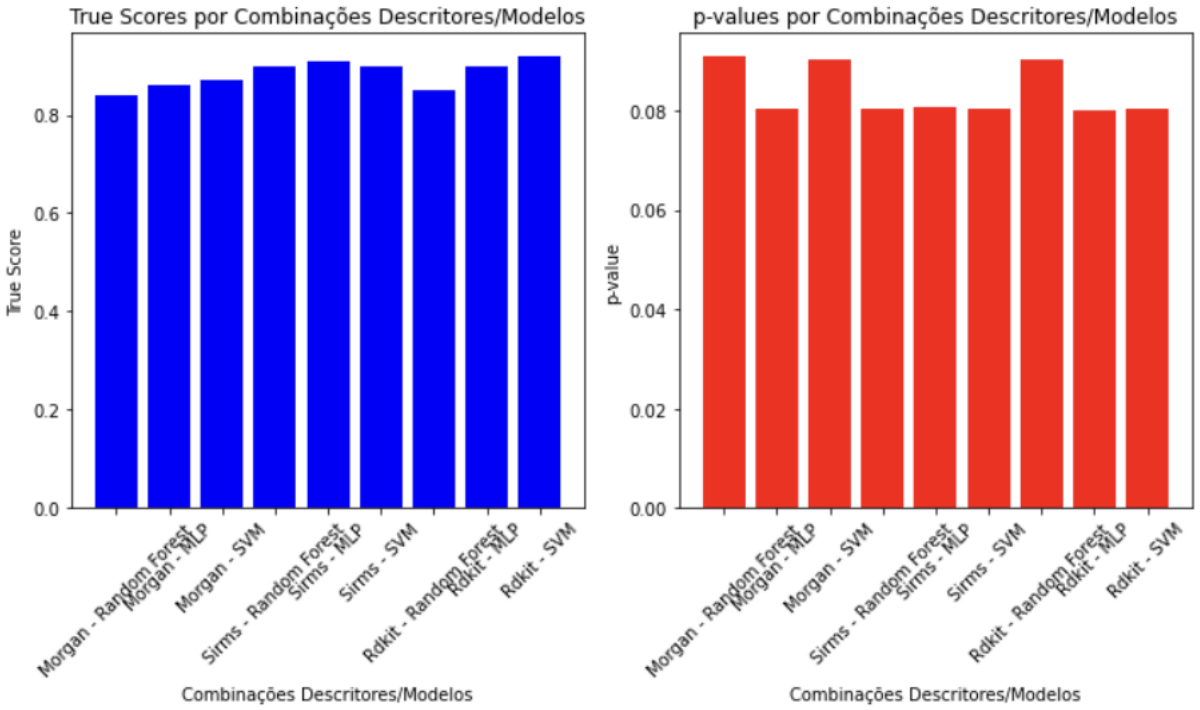
Esse teste consistiu em embaralhar (permutar) as etiquetas das amostras várias vezes (neste caso, 10 vezes) e calcular a métrica de avaliação (acurácia) para cada permutação (Tabela 17 e Figura 41). Essa ação resultou na criação de uma distribuição das pontuações de permutação, que representavam o desempenho por acaso. O p-valor foi calculado como a proporção de pontuações de permutação que eram iguais ou melhores do que o escore real.

Um p-valor muito pequeno, geralmente menor que 0,05, indicaria que o desempenho dos modelos com os dados reais era estatisticamente significativo em comparação com o desempenho aleatório. Portanto, o teste de permutação permitiu avaliar se os modelos apresentaram um desempenho estatisticamente significativo em relação aos dados reais, em comparação com os dados puramente aleatórios.

Tabela 17 – Teste de permutação

Descritores	Algoritmo	*True score*	Média per.	*p*-valor
Morgan	*Random Forest*	0,84	0,5	0.0910
Morgan	*Multilayer Perceptron*	0,86	0,5	0.0802
Morgan	SVM	0,87	0,5	0.0901
SiRMS	*Random Forest*	0,90	0,5	0.0802
SiRMS	*Multilayer Perceptron*	0,91	0,5	0.0808
SiRMS	SVM	0,90	0,5	0.0802
RDKit	*Random Forest*	0,85	0,5	0.0902
RDKit	*Multilayer Perceptron*	0,90	0,5	0.0801
RDKit	SVM	0,92	0,5	0.0803

Figura 41 – Gráficos que ilustram os testes de permutação. Fonte: Autoria própria.



Os descritores são uma ferramenta importante para a análise de dados químicos e biológicos, permitindo a representação de moléculas e substâncias de forma estruturada (XUE; BAJORATH, 2000). Neste estudo, analisamos o desempenho de três modelos de aprendizado de máquina diferentes, aplicando esses descritores:

- **Descritores Morgan:**
  - *Random Forest*: o modelo obteve um "*True score*" de 0,84, indicando um desempenho razoável. No entanto, destaca-se que o valor de "*p-value*" foi de 0,0910, sugerindo que esse resultado pode não ser estatisticamente significativo.

  - MLP: o modelo obteve um desempenho um pouco melhor, com um "*True score*" de 0,86. O valor de "*p-value*" foi de 0,0802, indicando uma melhora estatisticamente significativa em relação ao *Random Forest*.
  - SVM: o modelo obteve o melhor desempenho, com um "*True score*" de 0,87. No entanto, o valor de "*p-value*" foi de 0,0901, sugerindo que, apesar do desempenho superior, a diferença em relação ao *Random Forest* pode não ser estatisticamente significativa.

- **Descritores SiRMS:**
  - *Random Forest*: o modelo obteve um desempenho sólido, com um "*True score*" de 0,90. O valor de "*p-value*" foi de 0,0802, sugerindo que esse resultado é estatisticamente significativo.
  - MLP: o modelo também teve um desempenho muito bom, com um "*True score*" de 0,91. O valor de "*p-value*" foi de 0,0808, indicando uma melhoria estatisticamente significativa em relação ao *Random Forest*.
  - SVM: o modelo obteve um desempenho consistente, com um "*True score*" de 0,90. O valor de "*p-value*" foi de 0,0802, sugerindo que esse resultado é estatisticamente significativo.

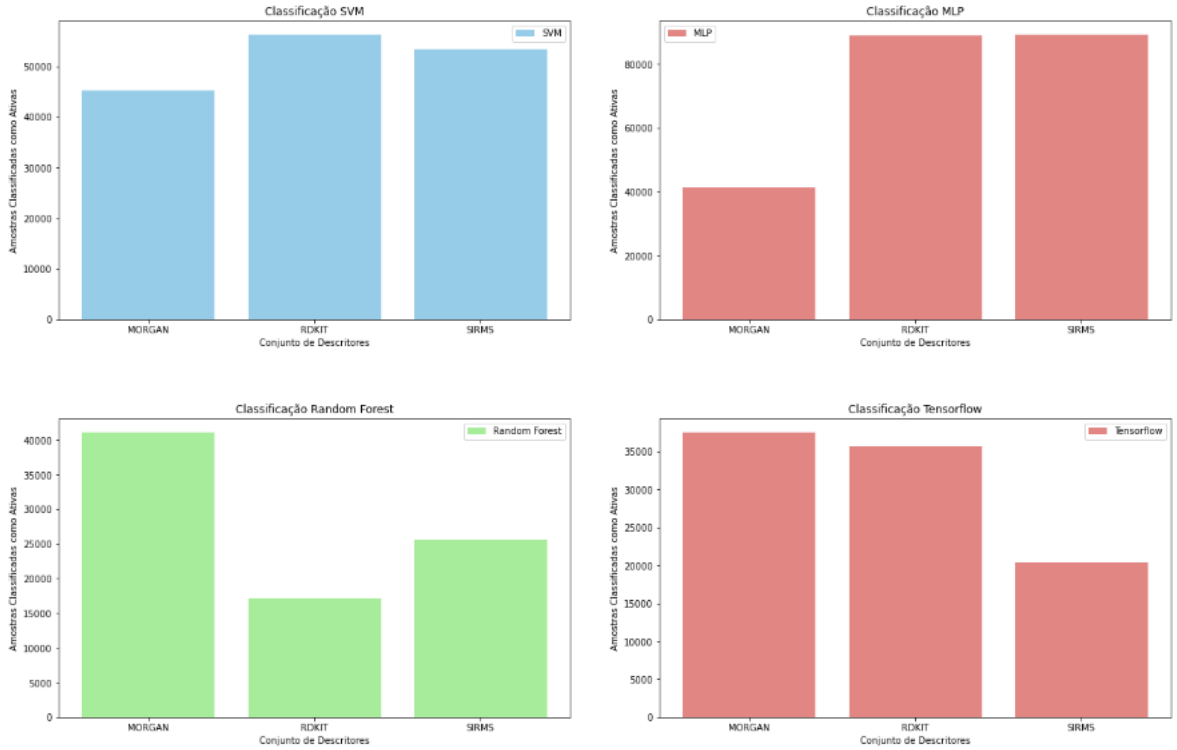
- **Descritores RDKit:**
  - *Random Forest*: o modelo teve um desempenho razoável, com um "*True score*" de 0,85. O valor de "*p-value*" foi de 0,0902, indicando que o resultado pode não ser estatisticamente significativo.
  - MLP: o modelo obteve um desempenho muito bom, com um "*True score*" de 0,90. O valor de "*p-value*" foi de 0,0801, indicando que esse resultado é estatisticamente significativo.
  - SVM: o modelo teve o melhor desempenho, com um "*True score*" de 0,92. O valor de "*p-value*" foi 0,0803, sugerindo que esse resultado é estatisticamente significativo.

5.4 Triagem virtual em bases de dados químicos

5.4.1 Execução do procedimento de triagem virtual

Para cada conjunto de descritores (Morgan, RDKit e SiRMS), quatro algoritmos de classificação foram aplicados (SVM, MLP, *Random Forest* e *TensorFlow* - Figura 42) em uma grande base de dados, composta por 101.097 amostras, obtidas da PubChem. Destaca-se que cada conjunto de descritores possuía diferentes dimensões, com 2048, 209 e 1764 características, respectivamente.

Figura 42 – Classificação usando o SVM, MLP, *Random Forest* e *TensorFlow*. Fonte: Autoria própria.



A Tabela 18 apresenta os valores considerados apenas quando estão dentro do domínio de aplicabilidade, com um nível de confiança superior ao limite de 70% (SUSHKO, 2011).

Tabela 18 – Consenso de modelos (número de compostos)

Conjunto de Descritores	SVM	MLP	*Random Forest*	*TensorFlow*	Consenso AD
Morgan	45.152	41.198	41.060	37.505	6.455
RDKit	56.229	89.058	17.183	35.748	3.773
SiRMS	53.447	89.156	25.636	20.438	3.629

- Em relação aos Descritores Morgan:
  - o SVM classificou 45.152 amostras como ativas dentro do domínio de aplicabilidade.
  - o MLP classificou 41.198 amostras como ativas dentro do domínio de aplicabilidade.
  - o *Random Forest* classificou 41.060 amostras como ativas dentro do domínio de aplicabilidade.
  - o *TensorFlow* classificou 37.505 amostras como ativas.

  - houve um consenso entre os modelos SVM, SVM AD, *Random Forest*, *Random Forest* AD, MLP e *TensorFlow*, com 6.455 amostras classificadas como ativas (Figura 43).

- Para os Descritores RDKit:

  - o SVM classificou 56.229 amostras como ativas dentro do domínio de aplicabilidade.

  - o MLP classificou 89.058 amostras como ativas dentro do domínio de aplicabilidade.

  - o *Random Forest* classificou 17.183 amostras como ativas dentro do domínio de aplicabilidade.

  - o *TensorFlow* classificou 35.748 amostras como ativas.

  - houve um consenso entre os modelos SVM, SVM AD, *Random Forest*, *Random Forest* AD, MLP e *TensorFlow*, com 3.773 amostras classificadas como ativas (Figura 43).

- Para os Descritores SiRMS:

  - o SVM classificou 53.447 amostras como ativas dentro do domínio de aplicabilidade.

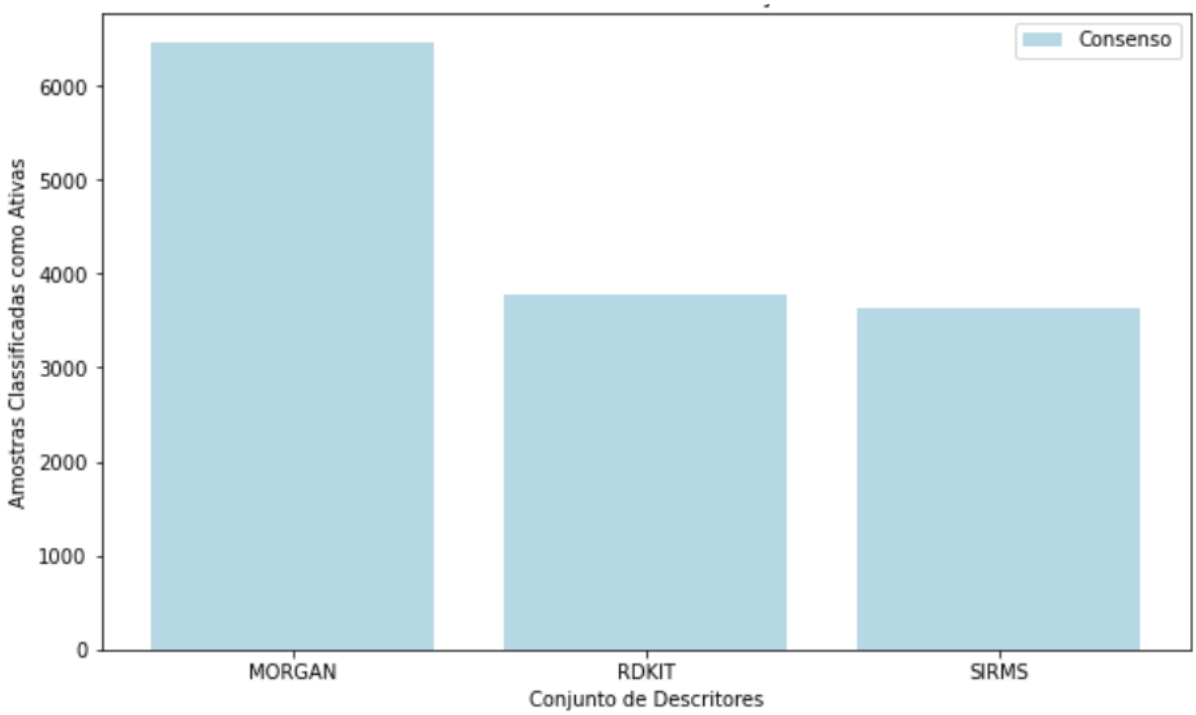
  - o MLP classificou 89.156 amostras como ativas dentro do domínio de aplicabilidade.

  - o *Random Forest* classificou 25.636 amostras como ativas dentro do domínio de aplicabilidade.

  - o *TensorFlow* classificou 20.438 amostras como ativas.

  - houve um consenso entre os modelos SVM, SVM AD, *Random Forest*, *Random Forest* AD, MLP e *TensorFlow*, com 3.629 amostras classificadas como ativas (Figura 43).

Figura 43 – Consenso de classificação. Fonte: Autoria própria.



5.4.2 Busca por similaridade

O *KNIME Analytics* foi utilizado para realizar a busca por similaridade (Figura 44) baseada em quatro compostos ativos (Figura 45) disponíveis na literatura, conforme detalhamento obtido do artigo de referência (GROSSBERG, 2003). Essa pesquisa foi realizada em uma base de dados altamente precisa, que continha um total de 117.379 compostos.

Figura 44 – Uso do KNIME Analytics para realizar a busca por similaridade. Fonte: Autoria própria.

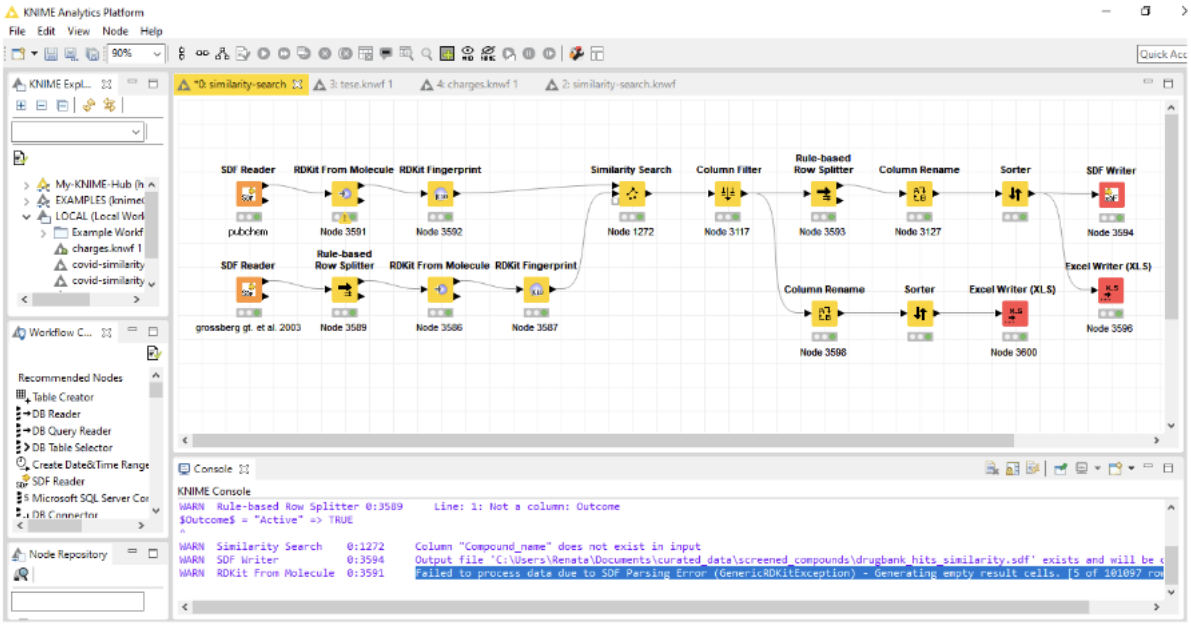
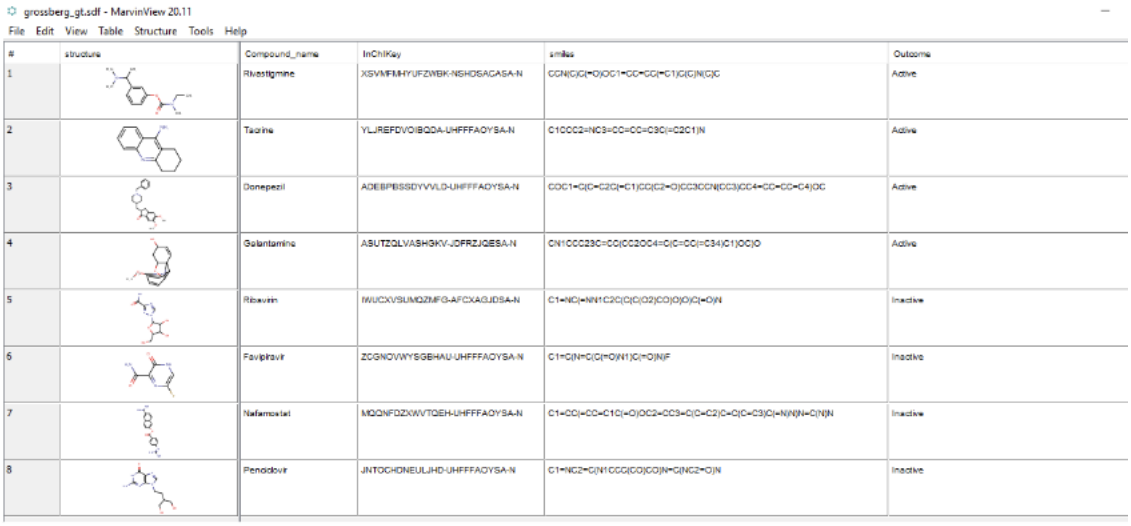


Figura 45 – Quatro compostos ativos disponíveis na literatura: Rivastigmine (PUBCHEM, 2023c), Tacrine (PUBCHEM, 2023d), Donepezil (PUBCHEM, 2023a) e Galantamine (PUBCHEM, 2023b). Fonte: Autoria própria.



Para realizar essa busca, foi utilizado o RDKit para extrair os descritores moleculares e, posteriormente, calcular o coeficiente de similaridade com base na distância usando a Similaridade de Tanimoto (MAGGIORA *et al.*, 2014), com um filtro de faixa de 0 a 0,9999999 para retornar as correspondências mais próximas. Como resultado final desse processo, um total de 5.837 compostos foram identificados e selecionados, os quais apresentaram uma similaridade significativa com os quatro compostos de referência.

5.4.3 Previsão de consenso dos compostos com os modelos obtidos

Após a busca por similaridade, os dados de Coeficiente de Tanimoto (*Similarity*) e do Vizinho Mais Próximo (*Nearest Neighbor*) foram mesclados nos resultados de consenso entre os descritores (Tabela 19). Esses resultados apresentam as informações sobre os compostos químicos mais similares identificados em relação ao composto de referência (Tacrine), utilizando o coeficiente de Tanimoto na busca por similaridade (MAGGIORA *et al.*, 2014). Cada linha representa um composto químico (identificado pelo CID - Chemical Identifier) e inclui a medida de similaridade em relação ao composto de referência, juntamente com a sua estrutura molecular simplificada (*Canonical*SMILES) e o composto mais semelhante encontrado (*Nearest Neighbor*). A similaridade varia de acordo com a estrutura molecular dos compostos, sendo que valores mais elevados indicam uma maior similaridade.

Tabela 19 – Consenso com a similaridade

ID	CID	*Canonical*SMILES	*Nearest Neighbor*	*Similarity (%)*
0	165748451	CC(C)(C)c1ccccc1C(=O)C(F)F	Donepezil	0.262530
1	126973612	CC(C)c1ccccc1C(=O)C(F)F	Donepezil	0.253333
2	118729284	CN(CCCCCCN1C(=O)c2ccccc2C1=O)Cc1ccccc1	Donepezil	0.314754
3	21994169	O=C1NC(=O)c2c(CCCN3CCC(Cc4ccccc4)CC3)cccc21	Donepezil	0.385714
4	22132546	Nc1ccc2c(c1)CN(CCCCN1C(=O)c3ccccc3C1=O)CC2	Donepezil	0.347181
5	12004040	c1ccc2ncc(Nc3ncnc4c3CCN(CC3CCCCC3)C4)cc2c1	Galantamine	0.412822
6	54542240	O=C1NC(=O)c2c(CCCN3CCc4ccccc4C3)cccc21	Donepezil	0.369673
7	54403061	O=C1NC(=O)c2c(CCCCN3CCc4ccccc4C3)cccc21	Donepezil	0.395980
8	60259671	CC1CCCN(Cc2ccc(CNC(=O)c3ccc4c(c3)C(=O)NC4=O)cc...	Donepezil	0.332971
9	119536775	O=C(NCCC1CCNC1)c1cccc(CN2C(=O)c3ccccc3C2=O)c1	Donepezil	0.322513
10	66587765	CCN(CC)c1cccc(-c2cc(C(=O)NC3CCCc4ccccc43)ccn2)c1	Galantamine	0.399876
11	22588138	CCCCC(CC)CNC(=O)CCCCCn1c(=O)[nH]c2ccccc2c1=O	Galantamine	0.326582
12	120179720	CNCCC1CCN(C(=O)c2ccc3c(c2)CCC(=O)N3)CC1	Donepezil	0.380859
13	17956492	Cc1cc(N(C)C(=O)NCCN2CCC(Cc3ccccc3)CC2)c2ccccc2n1	Tacrine	0.465433
14	647903	CCCCc1nc2ccccc2c(NC(=O)CN2CCN(C)CC2)c1CCC	Tacrine	0.714721
15	1099160	CCCc1nc2ccccc2c(NC(=O)CNC2CCCCC2)c1CC	Tacrine	0.674723
16	4218057	CCCc1nc2ccccc2c(NC(=O)C[NH+]2CCCCCC2)c1CC	Tacrine	0.675076
17	4990629	CCCc1nc2ccccc2c(NC(=O)C[NH+]2CCCCC2)c1CC	Tacrine	0.677126
18	6966754	CCCc1nc2ccccc2c(NC(=O)C[NH2+]C2CCCCC2)c1CC	Tacrine	0.674723
19	133412317	CCCc1cc(NCCC2CCN(C(C)=O)CC2)c2ccccc2n1	Tacrine	0.548712
20	55964361	CC1CC(C)CN(Cc2ccc(CNC(=O)C=Cc3cccnc3)cc2)C1	Donepezil	0.302455
21	119438861	CCNCc1ccccc1NC(=O)C1CCCN(C(=O)c2ccncc2)C1	Donepezil	0.331806
22	121108747	CCCc1ccc(C(=O)NCC2CCN(c3ccncc3)CC2)cc1	Donepezil	0.307607
23	14783862	Cc1ccccc1C(=O)NCCC1CCN(Cc2ccccc2)CC1	Donepezil	0.345588
24	56396266	CC1CC(C)CN(Cc2ccc(CNC(=O)c3cccc(F)c3F)cc2)C1	Donepezil	0.309979
25	38401175	Cc1ccc(F)cc1C(=O)NCc1ccc(CN2CCC(C)CC2)cc1	Donepezil	0.332594
26	46465375	Cc1ccccc1C(=O)NCC(=O)NCc1ccc(CN2CCCC(C)C2)cc1	Donepezil	0.331089
27	55714142	CNC(=O)c1ccc(C=CC(=O)NCc2ccccc2CN2CCCC(C)C2)cc1	Donepezil	0.329361
28	84422326	CN(C)CCCNC(=O)c1ccc(CNC(=O)c2ccc(C(C)(C)C)cc2)cc1	Donepezil	0.242925
29	95809500	O=C(c1ccncc1)N1CCC(c2cccc(Cc3ccccc3)n2)CC1	Galantamine	0.307245
30	95816347	Cc1cc(Cc2ccccc2)cc(C2CCCN(C(=O)c3ccncc3)C2)n1	Galantamine	0.348428
31	110249502	O=C(c1ccncc1)N1CCCC(c2cccc(Cc3ccccc3)n2)C1	Galantamine	0.346926
32	109236248	CC1CCCN(c2cncc(C(=O)NCCc3ccccc3F)c2)C1	Galantamine	0.317481
33	109227211	CC1CCN(c2cncc(C(=O)NCCc3ccccc3F)c2)CC1	Galantamine	0.311528
34	109103313	O=C(NCCC1=CCCCC1)c1cncc(C(=O)NCc2ccc(F)cc2)c1	Galantamine	0.255319
35	37027068	O=C(Nc1cccc(F)c1)C1CCN(C(=O)c2cccnc2)CC1	Donepezil	0.324111
36	46547516	O=C(NCc1cccnc1)C1CCCN(C(=O)Cc2ccccc2)C1	Donepezil	0.327818

Em seguida, foi realizado o consenso, selecionando dentre os 37 compostos, aqueles que apresentaram uma similaridade superior a 50% (0.50) - Tabela 20.

Tabela 20 – Consenso com a similaridade

CID	*Canonical*SMILES	*Nearest Neighbor*	*Similarity (%)*
647903	CCCCc1nc2ccccc2c(NC(=O)CN2CCN(C)CC2)c1CCC	Tacrine	0.714721
1099160	CCCc1nc2ccccc2c(NC(=O)CNC2CCCCC2)c1CC	Tacrine	0.674723
4218057	CCCc1nc2ccccc2c(NC(=O)C[NH+]2CCCCCC2)c1CC	Tacrine	0.675076
4990629	CCCc1nc2ccccc2c(NC(=O)C[NH+]2CCCCC2)c1CC	Tacrine	0.677126
6966754	CCCc1nc2ccccc2c(NC(=O)C[NH2+]C2CCCCC2)c1CC	Tacrine	0.674723
133412317	CCCc1cc(NCCC2CCN(C(C)=O)CC2)c2ccccc2n1	Tacrine	0.548712

Os resultados apresentam os *hits* finais obtidos por consenso entre quatro algoritmos e três descritores diferentes, alcançados por meio da busca por similaridade usando o Coeficiente de Tanimoto (Tabela 21). Cada *hit* foi assim detalhado:

- CID (*Chemical Identifier*):
  - o identificador único do composto químico.
- *Canonical*SMILES (*Simplified Molecular Input Line Entry System*):
  - uma representação simplificada da estrutura molecular do composto químico em formato de texto.
- *Nearest Neighbor* (Vizinho Mais Próximo):
  - o composto químico mais semelhante encontrado na busca por similaridade.
- Similaridade (Coeficiente de Tanimoto):
  - uma medida de quão similar o composto químico encontrado é em relação ao composto de referência. Quanto mais próximo de 1, maior a similaridade.

Abaixo estão alguns exemplos dos hits finais identificados (Figura 46):

- CID 647903:
  - *Canonical* SMILES: CCCCc1nc2ccccc2c(NC(=O)CN2CCN(C)CC2)c1CCC
  - *Nearest Neighbor*: Tacrine
  - Similaridade: 0.714721
  - Link PubChem: pubchem.ncbi.nlm.nih.gov/compound/647903
- CID 1099160:
  - *Canonical* SMILES: CCCc1nc2ccccc2c(NC(=O)CNC2CCCCC2)c1CC
  - *Nearest Neighbor*: Tacrine

  - Similaridade: 0.674723
  - Link PubChem: pubchem.ncbi.nlm.nih.gov/compound/1099160

- CID 4218057:
  - *Canonical* SMILES: CCCc1nc2ccccc2c(NC(=O)C[NH+]2CCCCCC2)c1CC
  - *Nearest Neighbor*: Tacrine
  - Similaridade: 0.675076
  - Link PubChem: pubchem.ncbi.nlm.nih.gov/compound/4218057

- CID 4990629:
  - *Canonical* SMILES: CCCc1nc2ccccc2c(NC(=O)C[NH+]2CCCCC2)c1CC
  - *Nearest Neighbor*: Tacrine
  - Similaridade: 0.677126
  - Link PubChem: pubchem.ncbi.nlm.nih.gov/compound/4990629

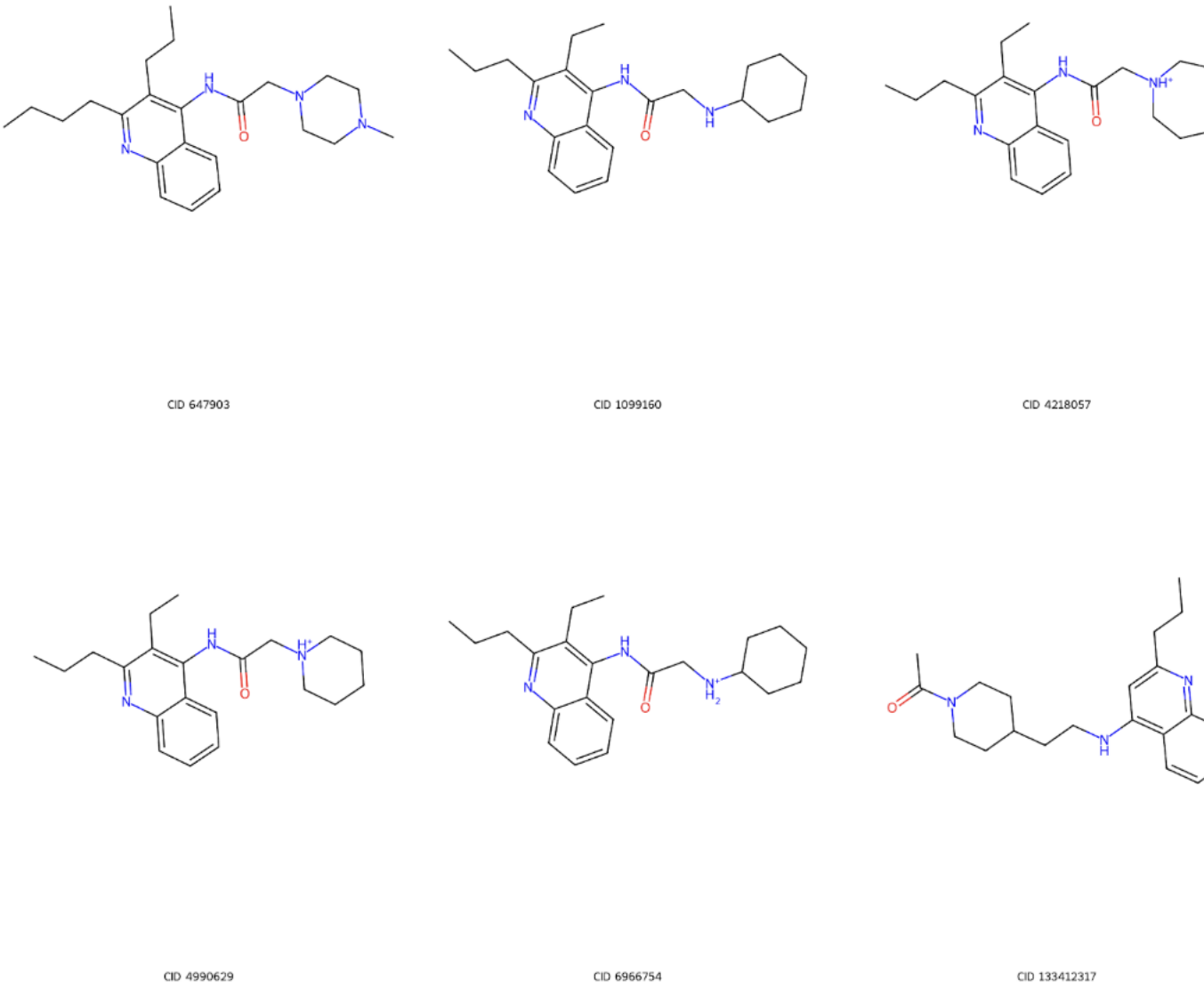
- CID 6966754:
  - *Canonical* SMILES: CCCc1nc2ccccc2c(NC(=O)C[NH2+]C2CCCCC2)c1CC
  - *Nearest Neighbor*: Tacrine
  - Similaridade: 0.674723
  - Link PubChem: pubchem.ncbi.nlm.nih.gov/compound/6966754

- CID 133412317:
  - *Canonical* SMILES: CCCc1cc(NCCC2CCN(C(C)=O)CC2)c2ccccc2n1
  - *Nearest Neighbor*: Tacrine
  - Similaridade: 0.548712
  - Link PubChem: pubchem.ncbi.nlm.nih.gov/compound/133412317

Tabela 21 – Consenso final (número de compostos)

Conjunto de Descritores	SVM	MLP	*Random Forest*	*TensorFlow*	Consenso AD	Consenso com Rigor	Consenso da Similaridade
Morgan	45.152	41.198	41.060	37.505	6.455	**37**	**6**
RDKit	56.229	89.058	17.183	35.748	3.773		
SiRMS	53.447	89.156	25.636	20.438	3.629		

Figura 46 – Compostos finais identificados após realizar a triagem virtual. Fonte: Autoria própria.



5.5 Discussões

Após uma análise dos resultados, destacam-se alguns pontos:

1. principais resultados:

    - significância estatística: os testes de permutação nos permitiram calcular o quão provável é que a performance dos modelos tenha sido alcançada por acaso. Essa probabilidade é representada pelos valores de p (*p-values*) associados a cada modelo e conjunto de descritores. Em geral, um *p-values* baixo (geralmente <0,05) indica que os resultados não são devido ao acaso.

- desempenho do modelo: observamos variações significativas no desempenho dos modelos em diferentes conjuntos de descritores. Por exemplo, o modelo SVM obteve uma alta acurácia com descritores RDKit, enquanto o MLP obteve uma alta acurácia com os descritores Morgan.
- domínio de aplicabilidade (AD): incorporamos o conceito de Domínio de Aplicabilidade (AD), que se refere à capacidade de um modelo fazer previsões confiáveis em uma determinada região do espaço de recursos. Essa avaliação foi realizada usando a técnica de validação cruzada com um limiar de AD. Os modelos que não atenderam ao limiar, foram considerados fora do domínio de aplicabilidade.

2. impacto do AD em relação aos algoritmos:

   - a inclusão do conceito de AD teve um impacto significativo nos resultados dos algoritmos, melhorando as métricas em uma média de 20-25%.
   - ao comparar os modelos com e sem AD, foi possível observar uma melhora substancial nas métricas, como Sensibilidade e Especificidade, com ganhos médios de 20-25%, demonstrando a importância do AD para melhorar a capacidade dos modelos em classificar de forma precisa as amostras dentro do seu domínio de aplicação (SUSHKO, 2011).
   - é importante destacar que a melhoria variou entre os algoritmos, sendo mais evidente nos modelos *Random Forest* e SVM, onde a inclusão do AD resultou em média de aprimoramento de 25-30%. Essa descoberta destaca ainda mais a relevância do AD como uma ferramenta essencial para otimizar o desempenho dos modelos em contextos específicos (BASKIN; KIREEVA; VARNEK, 2010).

3. desempenho do *TensorFlow* em relação aos algoritmos:

   - o *TensorFlow* também apresentou bons resultados, com desempenho semelhante ou superior em várias métricas em comparação com os modelos *Random Forest*, SVM e MLP, com ganhos médios de 10-15%.
   - esses resultados destacam que o *TensorFlow* é uma escolha robusta para as tarefas de classificação, fornecendo resultados competitivos em diversas métricas.

4. o benefício do AD em relação ao *TensorFlow*:

   - a inclusão do AD melhorou os resultados em termos de Sensibilidade e Especificidade em comparação com o *TensorFlow*, com ganhos médios de 15-20%.
   - esses resultados ressaltam que o AD desempenha um papel fundamental na melhoria da capacidade dos modelos de reconhecer amostras relevantes dentro do domínio de aplicabilidade.

Em síntese, a inserção do AD beneficiou, significativamente, o desempenho dos algoritmos de aprendizado de máquina, melhorando a capacidade de classificar amostras dentro do domínio de aplicabilidade com ganhos médios de 20-25% nas métricas relevantes (BASKIN; KIREEVA; VARNEK, 2010; ALAMRO *et al.*, 2023). Além disso, o *TensorFlow* se destacou como uma alternativa eficaz e competitiva em relação aos algoritmos tradicionais, demonstrando resultados consistentes com ganhos médios de 10-15% nas métricas. A escolha entre os modelos deve depender das métricas específicas mais relevantes para a aplicação, mas considerar o AD é crucial para melhorar a especificidade e a sensibilidade dos modelos.

5.5.1 Avaliação dos modelos em uma base de dados externa

Após treinar e validar os modelos, realizamos uma busca em uma grande base de dados (com 101.097 amostras) usando os modelos treinados. Essa avaliação resultou em algumas conclusões significativas:

- desempenho em grandes bases de dados: os modelos foram capazes de classificar com sucesso uma grande quantidade de amostras presentes na base de dados externa. Esse feito ressalta a capacidade dos modelos em lidar com conjuntos de dados de grande escala.

- resultados de consenso: além disso, realizamos o cálculo dos resultados de consenso entre os quatro algoritmos (SVM, MLP, *Random Forest* e *TensorFlow*) em três conjuntos de descritores diferentes. Essa abordagem nos permitiu identificar compostos químicos que foram classificados como ativos em consenso por todos os modelos, um procedimento importante para ressaltar as descobertas consistentes e confiáveis (ALAMRO *et al.*, 2023).

- busca por similaridade (Tanimoto): outro aspecto importante da avaliação foi a execução de busca por similaridade usando o coeficiente de Tanimoto para identificar compostos químicos semelhantes aos de referência (Tacrine). Essa abordagem desempenha um papel importante em aplicações voltadas para a descoberta de novos compostos farmacêuticos, ampliando o escopo das possibilidades de pesquisa (MAGGIORA *et al.*, 2014; GROSSBERG, 2003).

5.5.2 Implicações práticas e potencial de aplicação

Por fim, é importante discutir as implicações práticas desses resultados. Os modelos de *machine learning* e *deep learning* que foram treinados revelaram o seu valor quando aplicados a uma grande base de dados, proporcionando a capacidade de triagem de compostos químicos potencialmente ativos. Esse processo economiza tempo e recursos em

experimentos laboratoriais, priorizando compostos promissores para testes subsequentes (BAO *et al.*, 2023).

Além disso, a estratégia de busca por similaridade usando o coeficiente de Tanimoto é uma ferramenta importante para identificação de compostos químicos que compartilham características com um composto de referência, o que pode ser útil em pesquisa farmacêutica e química medicinal (MAGGIORA *et al.*, 2014; FERREIRA; ANDRICOPULO, 2018).

Portanto, este trabalho discutiu desde os testes de permutação para avaliação de modelos até a aplicação prática desses modelos em grandes bases de dados e busca por similaridade, destacando a relevância dessas técnicas na descoberta de novos compostos químicos com potencial atividade farmacológica.

6 CONCLUSÕES

Este trabalho abordou a aplicação de modelos de aprendizado de máquina e aprendizado profundo de máquina em três conjuntos de descritores diferentes (Morgan, RDKit e SiRMS) em uma grande base de dados químicos para classificação de amostras como ativas ou inativas dentro do domínio de aplicabilidade. Vários modelos, incluindo o SVM, MLP, *Random Forest* e *TensorFlow*, foram treinados e validados para cada conjunto de descritores.

As principais descobertas e conclusões deste estudo podem ser sintetizadas da seguinte forma:

- desempenho variável por conjunto de descritores: os modelos tiveram desempenhos variáveis em cada conjunto de descritores. Por exemplo, os descritores Morgan resultaram em um menor número de amostras classificadas como ativas, enquanto os descritores RDKit tiveram um número maior de amostras ativas.

- diferenças nos modelos: cada modelo apresentou desempenho diferente para cada conjunto de descritores, ressaltando a importância da seleção adequada de modelos para conjuntos de descritores específicos.

- consenso entre modelos: foi observado que, em todos os conjuntos de descritores, um número significativo de amostras foi classificado como ativas em consenso por todos os modelos, sugerindo a robustez dessas amostras e sua importância.

- importância da escolha de descritores: a escolha adequada de descritores revelou-se crítica para o desempenho dos modelos, uma vez que diferentes conjuntos de descritores capturam informações químicas de maneira única, resultando em diferentes resultados.

- potencial de aplicações futuras: os modelos treinados e os resultados obtidos têm potencial de aplicação em triagem de compostos químicos, descoberta de medicamentos e pesquisa farmacêutica, onde a classificação precisa de compostos como ativos ou inativos é fundamental.

- necessidade de validação externa: embora os modelos tenham demonstrado bons desempenhos nos dados de validação interna, a validação externa em conjuntos de dados independentes foi essencial para avaliar verdadeiramente a robustez dos modelos.

- considerações éticas e de segurança: a aplicação desses modelos na indústria farmacêutica e química deve ser realizada com considerações éticas e de segurança, garantindo que os compostos identificados como ativos sejam seguros e eficazes.

Portanto, esta análise demonstrou que modelos de aprendizado de máquina e aprendizado profundo de máquina têm o potencial de melhorar a triagem e a classificação de compostos químicos em grandes bases de dados. No entanto, a escolha criteriosa dos descritores e modelos é fundamental, assim como a validação externa é necessária antes da aplicação prática. A pesquisa continuada nesse campo visa aprimorar ainda mais a precisão e a eficácia dos modelos, contribuindo para avanços significativos nas áreas de química e farmacologia.

REFERÊNCIAS

ABADI, M. *et al.* Tensorflow: Large-scale machine learning on heterogeneous distributed systems. **Distributed, Parallel, and Cluster Computing**, v. 1, 2016.

ALAMRO, H. *et al.* Exploiting machine learning models to identify novel alzheimer's disease biomarkers and potential targets. **Scientific Reports**, n. 13, p. 4979, 2023.

ALVES, V. *et al.* Qsar modeling of sars-cov mpro inhibitors identifies sufugolix, cenicriviroc, proglumetacin, and other drugs as candidates for repurposing against sars-cov-2. **Molecular Informatics**, v. 40, n. 1, p. 2000113, 2021.

ALVES, V. M. *et al.* Cheminformatics: an introduction. **Química NOva**, v. 41, n. 2, p. 202–212, 2018.

ALZUBAIDI, L. *et al.* Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. **Journal of Big Data**, v. 8, n. 1, p. 53, 2021.

BAO, L. Q. *et al.* Development of activity rules and chemical fragment design for in silico discovery of ache and bace1 dual inhibitors against alzheimer's disease. **Molecules**, v. 28, n. 8, p. 3588, 2023.

BASKIN, I.; KIREEVA, N.; VARNEK, A. The one-class classification approach to data description and to models applicability domain. **Mol Inform**, v. 29, p. 8–9, 2010.

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. **Journal of Machine Learning Research**, v. 13, p. 281–305, 2012.

BERMAN, H.; HENRICK, K.; NAKAMURA, H. Announcing the worldwide protein data bank. **Nature Structural and Molecular Biology**, v. 10, n. 12, p. 980, 2003.

BEY, R. *et al.* Fold-stratified cross-validation for unbiased and privacy-preserving federated learning. **Journal of the American Medical Informatics Association**, v. 27, n. 8, p. 1244–1251, 2020.

BOBROWSKI, T. *et al.* Computational models identify several fda approved or experimental drugs as putative agents against sars-cov-2. **ChemRxiv**, 2020.

BREIMAN, L. Random forests. **Machine Learning**, v. 45, p. 5–32, 2001.

BRUCE, R. Semi-supervised learning using prior probabilities and em. **IJCAI Workshop on Text Learning: Beyond Supervision**, Springer, p. 1–23, 2001.

BUNIN, B. *et al.* Chemoinformatics theory. *In*: _____. **Chemoinformatics**. [*S.l.: s.n.*]: Springer, Dordrecht, 2007.

CARPENTER, K. A. *et al.* Deep learning and virtual drug screening. **Future Medicinal Chemistry**, v. 10, n. 21, p. 2557–2567, 2018.

CARPENTER, K. A.; HUANG, X. Machine learning-based virtual screening and its applications to alzheimer's drug discovery: A review. **Current Pharmaceutical Design**, v. 24, n. 28, p. 3347–3358, 2018.

CHAMJANGALI, M. A. Modelling of cytotoxicity data (cc50) of anti-hiv 1-[5-chlorophenyl) sulfonyl]-1h-pyrrole derivatives using calculated molecular descriptors and levenberg-marquardt artificial neural network. **Chemical Biology and Drug Design**, v. 73, n. 4, p. 456–465, 2020.

CHEIRDARIS, D. G. Artificial neural networks in computer-aided drug design: An overview of recent advances. **Advances in Experimental Medicine and Biology**, Vlamos, P. (eds) GeNeDis 2018, v. 1194, 2020.

CHEMBL. **ChEMBL web services API live documentation Explorer**. 1a. ed. [*S.l.: s.n.*]: Web Services specification, 2023. <https://www.ebi.ac.uk/chembl/api/data/docs>.

CHEN, H.; KOGEJ, T.; ENGKVIST, O. Cheminformatics in drug discovery, an industrial perspective. **Molecular informatics**, v. 37, n. 9-10, p. e1800041, 2018.

CHEN, R. *et al.* Machine learning for drug-target interaction prediction. **Molecules**, v. 23, n. 9, p. 2208–2223, 2018.

CHERKASOV, A. *et al.* Qsar modeling: where have you been? where are you going to? **Journal of Medicinal Chemistry**, v. 57, n. 12, p. 4977–5010, 2014.

DAI, R. *et al.* Anti-alzheimer's disease potential of traditional chinese medicinal herbs as inhibitors of bace1 and ache enzymes. **Biomed Pharmacother**, n. 154, p. 113576, 2022.

DAS, S.; CHAKRABORTY, S.; BASUCORRESPONDING, S. Hybrid approach to sieve out natural compounds against dual targets in alzheimer's disease. **Scientific Reports**, v. 9, p. 3714, 2019.

DELANOGARE, E. *et al.* Hipótese amiloide e o tratamento da doença de alzheimer: revisão dos estudos clínicos realizados. **VITTALLE - Revista De Ciências Da Saúde**, v. 31, n. 1, p. 84–106, 2019.

DHAMODHARAN, G.; MOHAN, C. G. Machine learning models for predicting the activity of ache and bace1 dual inhibitors for the treatment of alzheimer's disease. **Molecular Diversity**, n. 26, p. 1501–1517, 2022.

DOBCHEV, D.; KARELSON, M. Have artificial neural networks met expectations in drug discovery as implemented in qsar framework? **Expert Opinion on Drug Discovery**, v. 11, n. 7, p. 627–39, 2016.

EMBL-EBI. **ChEMBL - European Molecular Biology Laboratory**. 2020. Available at: <https://www.ebi.ac.uk/chembl/>.

FARA, D.; A.L.; OPREA, T. **Cheminformatics - Basics: 2D and 3D Molecular Structures**. 1a. ed. [*S.l.: s.n.*]: Chemical Structure Drawing Packages, 2019.

FAST, E.; CHEN, B. Potential t-cell and b-cell epitopes of 2019-ncov. **bioRxiv**, Cold Spring Harbor Laboratory, p. 1–13, 2020.

FAWCETT, T. An introduction to roc analysis. **Pattern Recognition Letters**, v. 27, p. 861–874, 2006.

FERREIRA, L.; ANDRICOPULO, A. Quimioinformática e aprendizado de máquinas: Um olhar sobre a pesquisa e o desenvolvimento de fármacos. **Química e Derivados**, p. 42–44, 2018.

FEURER, M.; HUTTER, F. The springer series on challenges in machine learning. *In*: _____. **Automated Machine Learning**. [*S.l.: s.n.*]: Springer, 2019. cap. Hyperparameter Optimization, p. 3–33.

FIGUERAS, J. Morgan revisited. **Journal of Chemical Information and Computer Sciences**, v. 33, n. 5, p. 717–718, 1993.

FOURCHES, D. Cheminformatics: At the crossroad of eras. *In*: _____. **Application of Computational Techniques in Pharmacy and Medicine**. [*S.l.: s.n.*]: Springer, Dordrecht, 2014.

FOURCHES, D.; MURATOV, E.; TROPSHA, A. Trust, but verify: On the importance of chemical structure curation in cheminformatics and qsar modeling research. **Journal of Chemical Information and Modeling**, v. 50, n. 7, p. 1189–1204, 2010.

FOURCHES, D.; MURATOV, E.; TROPSHA, A. Trust, but verify ii: A practical guide to chemogenomics data curation. **Journal of Chemical Information and Modeling**, v. 56, n. 7, p. 1243–1252, 2016.

FRIEDMAN, J. **Another approach to polychotomous classification**. [*S.l.*], 1996. Available at: <http://www-stat.stanford.edu/~jhf/ftp/poly.ps.Z>.

GARDNER, M. W.; DORLING, S. R. Artificial neural networks (the multilayer perceptron) — a review of applications in the atmospheric science. **Atmospheric Environment**, v. 32, n. 14, p. 2627–2636, 1998.

GERTRUDES, J. *et al.* Machine learning techniques and drug design. **Current Medicinal Chemistry**, v. 19, n. 25, p. 4289 – 4297, 2012.

GIL, A. C. **Como elaborar projetos de pesquisa**. 6a. ed. [*S.l.: s.n.*]: Editora Atlas, 2017.

GOLBRAIKH, A.; TROPSHA, A. Qsar modeling using chirality descriptors derived from molecular topology. **Journal of Chemical Information and Modeling**, v. 43, n. 1, p. 144–154, 2003.

GONCALVES, A. **Máquina de Vetores Suporte**. [*S.l.*], 2008. Available at: <https://andreric.github.io/files/pdfs/svm.pdf>.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. 1a. ed. [*S.l.: s.n.*]: MIT Press, 2016. <http://www.deeplearningbook.org>.

GOOGLE, B. **About the Google Brain Team**. 2023. Available at: <https://research.google/teams/brain/>.

GORB, L.; KUZ'MIN, V.; MURATOV, E. **Application of Computational Techniques in Pharmacy and Medicine**. 1a. ed. [*S.l.: s.n.*]: Springer Science, 2014.

GROSSBERG, G. T. Cholinesterase inhibitors for the treatment of alzheimer's disease:: getting on and staying on. **Current therapeutic research, clinical and experimental**, v. 64, n. 4, p. 216–35, 2003.

GUHA, R.; DRIE, J. Structure–activity landscape index: identifying and quantifying activity cliffs. **Journal of Chemical Information and Modeling**, v. 448, n. 3, p. 646–58, 2008.

GULIA, A.; DHAIYA, A.; ANSHUL. A review paper on support vector machine. **Global Journal of Engineering Science and Researches**, v. 6, n. 6, p. 313–318, 2019.

GUPTA, R. *et al.* Artificial intelligence to deep learning: machine intelligence approach for drug discovery. **Molecular diversity**, v. 25, n. 3, p. 1315–1360, 2021.

GUY, R. *et al.* Rapid repurposing of drugs for covid-19. **Science**, v. 368, n. 6493, p. 829–830, 2020.

HAYKIN, S. S. **Redes neurais: princípios e prática**. 2a. ed. [*S.l.: s.n.*]: Bookman, 2001.

HAYKIN, S. S. **Neural Networks and Learning Machines**. 3a. ed. [*S.l.: s.n.*]: New Jersey: Prentice Hall, 2009.

HORVATH, T.; ALDAHDOOH, J. **Investigating the importance of meta-features for classification tasks in meta-learning**. 2017. Tese (Doutorado) — Faculty of Informatics. Eötvös Loránd University. Budapest, 2017.

HU, Y. *et al.* Identify compounds' target against alzheimer's disease based on in-silico approach. **Current Alzheimer Research**, v. 16, n. 3, p. 193–208, 2019.

INCHITRUST. **InChI Trust - InChI: open-source chemical structure representation algorithm**. 2020. Available at: <https://www.inchi-trust.org/technical-faq-2/>.

IQBAL, S. *et al.* On the analyses of medical images using traditional machine learning techniques and convolutional neural networks. **Archives of computational methods in engineering : state of the art reviews**, v. 30, n. 5, p. 3173–3233, 2023.

JAMES, G. *et al.* **An Introduction to Statistical Learning with Applications in R**. 8a. ed. [*S.l.: s.n.*]: Springer Texts in Statistics, 2017.

JANG, C. *et al.* Identification of novel acetylcholinesterase inhibitors designed by pharmacophore-based virtual screening, molecular docking and bioassay. **Scientific reports**, n. 8, p. 14921, 2018.

JOHNSON, R. B.; ONWUEGBUZIE, A. J.; TURNER, L. A. Toward a definition of mixed methods research. **Journal of Mixed Methods Research**, v. 1, n. 2, p. 112–133, 2007.

JORNER, K. *et al.* Machine learning meets mechanistic modelling for accurate prediction of experimental activation energies. **Chemical Science**, The Royal Society of Chemistry, v. 12, n. 3, p. 1163–1175, 2021.

JUNG, Y. Multiple predicting k-fold cross-validation for model selection. **Journal of Nonparametric Statistics**, v. 30, n. 1, p. 197–215, 2018.

KANEHISA, M. *et al.* New approach for understanding genome variations in kegg. **Nucleic Acids Research**, v. 47, n. D1, p. D590–D595, 2019.

KARELSON, M. *et al.* Correlation of blood-brain penetration and human serum albumin binding with theoretical descriptors. **Archive for Organic Chemistry**, v. 2008, n. 16, p. 38–60, 2008.

KNIME. **KNIME Analytics Platform**. 2021. Available at: <https://www.knime.com/knime-analytics-platform>.

KULKARNI, A.; CHONG, D.; BATARSEH, F. Statistical assessment metrics. *In*: _____. **Data Democracy**. [*S.l.: s.n.*]: Academic Press, 2020. cap. 5 - Foundations of data imbalance and solutions for a data democracy, p. 83–106. ISBN 978-0-12-818366-3.

KUMAR, V.; KRISHNA, S.; SIDDIQI, M. Virtual screening strategies: Recent advances in the identification and design of anti-cancer agents. **Methods**, v. 71, p. 64–70, 2015.

LIASHCHYNSKYI, P.; LIASHCHYNSKYI, P. Grid search, random search, genetic algorithm: A big comparison for nas. **arXiv**, v. 1912.06059, p. 1–11, 2019.

LIPINSKI, C. A. *et al.* Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. **Advanced Drug Delivery Reviews**, v. 23, n. 1-3, p. 3–25, 1997.

LO, Y.-C. *et al.* Machine learning in chemoinformatics and drug discovery. **Drug Discovery Today**, v. 23, n. 8, p. 1538–1546, 2018.

MAGGIORA, G. *et al.* Molecular similarity in medicinal chemistry. **Journal of Medicinal Chemistry**, v. 57, n. 8, p. 3186–204, 2014.

MAGGIORA, G. M. On outliers and activity cliffs–why qsar often disappoints. **Journal of Chemical Information and Modeling**, v. 45, n. 4, p. 1535, 2006.

MAZZOLARI, A.; VISTOLI, G. **In silico approaches in drug design and development: applications to rational ligand design and metabolism prediction**. 2015. Dissertação (Mestrado) — Università Degli Studi Di Milano. Department of Pharmaceutical Sciences, 2015.

MITCHELL, T. **Machine Learning**. 1a. ed. [*S.l.: s.n.*]: McGraw-Hill Science, 1997.

MOUCHLIS, V. D. *et al.* Computer-aided drug design of b-secretase, y-secretase and anti-tau inhibitors for the discovery of novel alzheimer's therapeutics. **International Journal of Molecular Sciences**, v. 21, n. 3, p. 703, 2020.

NEVES, B. J. *et al.* Qsar-based virtual screening: Advances and applications in drug discovery. **Frontiers in Pharmacology**, v. 9, p. 1275–1282, 2018.

NICKEL, J. *et al.* Superpred: Update on drug classification and target prediction. **Nucleic Acids Research**, v. 42, p. W26–31, 2014.

NIH, N. I. of H. **PubChem**. 2020. Available at: <https://pubchem.ncbi.nlm.nih.gov/>.

OJALA, M.; GARRIGA, G. C. Permutation tests for studying classifier performance. **Journal of Machine Learning Research**, v. 11, p. 1833–1863, 2010.

ORGANIZATION, W. H. **Global status report on the public health response to dementia**. [*S.l.*], 2021. Available at: <https://apps.who.int/iris/bitstream/handle/10665/344701/9789240033245-eng.pdf>.

PANOV, P.; DZEROSKI, S. Combining bagging and random subspaces to create better ensembles. **Advances in Intelligent Data Analysis VII, 7th International Symposium on Intelligent Data Analysis**, p. 118–129, 2007.

PANTELEEV, J.; GAO, H.; JIA, L. Recent applications of machine learning in medicinal chemistry. **Bioorganic and Medicinal Chemistry Letters**, v. 28, n. 17, p. 2807–2815, 2018.

PARVANDEH, S. *et al.* Consensus features nested cross-validation. **Bioinformatics**, v. 36, n. 10, p. 3093–3098, 2020.

PATEL, L. *et al.* Machine learning methods in drug discovery. **Molecules**, v. 25, n. 22, p. 5277, 2020.

PEDREGOSA, F. Hyperparameter optimization with approximate gradient. **Proceedings of the International conference on Machine Learning (ICML)**, v. 1, n. 5, p. 1–14, 2016.

PEDREGOSA, F. *et al.* Scikit-learn: Machine learning in python. **Journal of Machine Learning Research**, v. 12, n. 85, p. 2825–2830, 2011. Available at: <http://jmlr.org/papers/v12/pedregosa11a.html>.

PUBCHEM. **Donepezil**. [*S.l.*], 2023. Available at: <https://pubchem.ncbi.nlm.nih.gov/compound/Donepezil>.

PUBCHEM. **Galantamine**. [*S.l.*], 2023. Available at: <https://pubchem.ncbi.nlm.nih.gov/compound/Galantamine>.

PUBCHEM. **Rivastigmine**. [*S.l.*], 2023. Available at: <https://pubchem.ncbi.nlm.nih.gov/compound/Rivastigmine>.

PUBCHEM. **Tacrine**. [*S.l.*], 2023. Available at: <https://pubchem.ncbi.nlm.nih.gov/compound/Tacrine>.

PUZYN, T.; LESZCZYNSKI, J.; CRONIN, M. **Recent Advances in QSAR Studies - Methods and Applications**. 8a. ed. [*S.l.: s.n.*]: Springer Netherlands, 2010.

QUADRI, T. W. *et al.* Multilayer perceptron neural network-based qsar models for the assessment and prediction of corrosion inhibition performances of ionic liquids. **Computational Materials Science**, v. 214, 2022.

RANDHAWA, G. *et al.* Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: Covid-19 case study. **PLoS One**, v. 15, n. 4, p. e0232391, 2020.

RAO, A.; VAZQUEZ, J. Identification of covid-19 can be quicker through artificial intelligence framework using a mobile phone–based survey when cities and towns are under quarantine. **Infection Control and Hospital Epidemiology**, v. 41, n. 7, p. 826–830, 2020.

RAO, H. *et al.* Prediction of hiv-1 protease inhibitors using machine learning approaches. **QSAR and Combinatorial Science**, v. 28, n. 11-12, p. 1346–1357, 2009.

RASHDAN, H. R. M.; ABDELMONSEF, A. H. In silico study to identify novel potential thiadiazole-based molecules as anti-covid-19 candidates by hierarchical virtual screening and molecular dynamics simulations. **Springer Nature**, v. 33, n. 5, p. 1727–1739, 2022.

RDKIT. **rdkit.Chem.Lipinski module - Calculation of Lipinski parameters for molecules**. 2023. Available at: <https://www.rdkit.org/docs/source/rdkit.Chem.Lipinski.html>.

SAKAI, M. *et al.* Prediction of pharmacological activities from chemical structures with graph convolutional neural networks. **Scientific Reports**, v. 11, n. 525, 2021.

SHARMA, S.; SHARMA, D. Intelligently applying artificial intelligence in chemoinformatics. **Current Topics in Medicinal Chemistry**, v. 18, n. 20, p. 1804–1826, 2018.

SUSHKO, I. **Applicability domain of QSAR models**. 2011. Dissertação (Mestrado) — Chair of Genome-Oriented Bioinformatics. Technical University of Munich. Ludwig Maximilian University of Munich, 2011.

SVETNIK, V. *et al.* Random forest: A classification and regression tool for compound classification and qsar modeling. **Journal of Chemical Information and Computer Sciences**, v. 43, n. 6, p. 1947–1958, 2003.

SZKLARCZYK, D. *et al.* Stitch 5: Augmenting protein-chemical interaction networks with tissue and affinity data. v. 44, n. D1, p. D380–4, 2016.

TALLON-BALLESTEROS, A.; RIQUELME, J. Data mining methods applied to a digital forensics task for supervised machine learning. **Computational Intelligence in Digital Forensics: Forensic Investigation and Applications**, Studies in Computational Intelligence, n. 555, p. 413–428, 2014.

TEJERA, E. *et al.* Drugs repurposing using qsar, docking and molecular dynamics for possible inhibitors of the sars-cov-2 mpro protease. **Molecules**, v. 25, n. 21, p. 5172, 2020.

TENSORFLOW, C. **Introdução ao TensorFlow**. 2023. Available at: <https://www.tensorflow.org/learn?hl=pt-br>.

TENSORFLOW, C. **tf.keras.Sequential**. 2023. Available at: <https://www.tensorflow.org/api_docs/python/tf/keras/Sequential>.

TODESCHINI, R.; CONSONNI, V. **Handbook of Molecular Descriptors**. 1a. ed. [*S.l.: s.n.*]: Wiley-VCH, 2000.

TROPSHA, A. Best practices for qsar model development, validation, and exploitation. **Molecular Informatics**, v. 29, n. 6-7, p. 1868–1743, 2010.

TROPSHA, A. *et al.* Predictive qsar modeling: Methods and applications in drug discovery and chemical risk assessment. **Handbook of Computational Chemistry**, Springer, p. 1–48, 2017.

ULRICH, E. L. *et al.* Biomagresbank. **Nucleic Acids Research**, v. 36, p. D402–D408, 2008.

VIEIRA, U. K. S. M.; SOUSA, J. M. C. Cohen's kappa coefficient as a performance measure for feature selection. **International Conference on Fuzzy Systems**, p. 1–8, 2010.

WALCZAK-NOWICKA, L. J.; HERBET, M. Acetylcholinesterase inhibitors in the treatment of neurodegenerative diseases and the role of acetylcholinesterase in their pathogenesis. **International journal of molecular sciences**, v. 22, n. 17, p. 9290, 2021.

WILLETT, P. Similarity-based virtual screening using 2d fingerprints. **Drug Discovery Today**, v. 11, n. 23-24, p. 1046–53, 2006.

WISHART, D. *et al.* Drugbank 5.0: A major update to the drugbank database for 2018. **Nucleic Acids Research**, v. 46, p. D1074–D1082, 2018.

WORACHARTCHEEWAN, A. *et al.* Modeling the activity of furin inhibitors using artificial neural network. **European Journal of Medicinal Chemistry**, v. 44, n. 4, p. 1664–1673, 2009.

WU, J. *et al.* Hyperparameter optimization for machine learning models based on bayesian optimization. **Journal of Electronic Science and Technology**, v. 17, n. 1, 2019.

WYNANTS, L. *et al.* Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal. **BMJ**, v. 369, p. m1328, 2020.

XUE, L.; BAJORATH, J. Molecular descriptors in chemoinformatics, computational combinatorial chemistry, and virtual screening. **Combinatorial Chemistry and High Throughput Screening**, v. 3, n. 363, 2000.

YANG, X. *et al.* The one-against-all partition based binary tree support vector machine algorithms for multi-class classification. **Neurocomputing**, v. 113, p. 1–7, 2013.

ZHU, H. *et al.* Big data in chemical toxicity research: The use of high-throughput screening assays to identify potential toxicants. **Chemical Research in Toxicology**, v. 27, n. 10, p. 1643–1651, 2014.