SINERGIA ENTRE APRENDIZADO DE MÁQUINA E DINÂMICA MOLECULAR PARA PREDIÇÃO DE PROPRIEDADES MOLECULARES: UMA REVISÃO DA LITERATURA

Felipe Cavalcanti de Godoy[a] e Luiz Filipe Tsarbopoulos de Resende[b,*]

[a] Departamento de Química Fundamental da Universidade de São Paulo (USP), São Paulo - SP, Brasil

[b] Departamento de Bioquímica e Biologia Molecular, Escola de Artes, Ciências e Humanidades da Universidade de São Paulo (USP), São Paulo - SP, Brasil.

*e-mail: tsarbopoulos@usp.br

Resumo: Este artigo revisa a aplicação de técnicas de Aprendizado de Máquina (*Machine Learning* - ML) na Dinâmica Molecular (*Molecular Dynamics* - MD) para a predição de propriedades moleculares. Inicialmente, são discutidos os conceitos fundamentais de MD e ML, para então evidenciar os pontos sinérgicos entre ML e MD principalmente na redução da demanda computacional e o aumento da precisão das predições nas simulações. Além disso, são descritas as principais áreas de aplicação da MD que se beneficiam da integração de ML, com destaque para a predição de propriedades de biomoléculas, materiais e catálise. Em ML são apresentados os algoritmos mais comuns como kNN, k-Means Clustering (kMC), SVM, Random Forests, Redes Neurais Artificiais, Regressão Logística e Análise de Componentes Principais (PCA), destacando suas vantagens e limitações de aplicações. Por fim, o artigo aborda os principais desafios da aplicação de ML na MD, como a necessidade de grandes volumes de dados de alta qualidade, a questão da interpretabilidade dos modelos complexos e as estratégias para mitigar problemas como o *overfitting*, oferecendo perspectivas futuras promissoras para o campo.

Palavras-chave: Aprendizado de Máquina; Dinâmica Molecular; Propriedades Moleculares; Predição; Modelagem; Algoritmo.

The synergy between machine learning and molecular dynamics for predicting molecular properties: A literature review

Abstract: This paper reviews the application of Machine Learning (ML) techniques in Molecular Dynamics (MD) for predicting molecular properties. Initially the fundamental concepts of MD and ML are discussed highlighting the synergic points between ML and MD in reducing computational demand and predictions accuracy improvement in simulations. Moreover, the main MD application areas that benefit from ML integration are described, focusing on the prediction of biomolecular, material, and catalytic properties. In ML, the most commonly used algorithms are presented including kNN, k-Means Clustering (kMC), SVM, Random Forests, Artificial Neural Networks, Logistic Regression, and Principal Component Analysis (PCA), highlighting their advantages and application limitations. Finally, the paper addresses the main challenges of applying ML in MD such as the need for large volumes of high-quality data, interpretability issues of complex models and strategies to mitigate problems like *overfitting*, providing promising future perspectives for the field.

Keywords: Machine Learning; Molecular Dynamics; Molecular Properties; Prediction; Modeling; Algorithm.

1. INTRODUÇÃO

A interpretação do mundo natural por meio de uma perspectiva matemática tem sido uma característica marcante nos estudos das ciências naturais. Historicamente, o desenvolvimento e a aplicação de recursos matemáticos, aliados ao caráter inventivo dos modelos teóricos, aumentaram significativamente a complexidade desses modelos em diversas áreas do conhecimento.[1] Nesse contexto, a matemática atua como uma base sólida na qual os modelos teóricos se ancoram para representar o mundo natural.

Com o avanço exponencial da tecnologia, o mundo digital, fundamentado em princípios matemáticos, apresenta novas perspectivas para a elaboração, validação e representação de modelos teóricos, superando as expectativas de pesquisadores do passado. Entre os recursos disponibilizados pelo desenvolvimento computacional, destaca-se a técnica da Molecular Dynamics (MD).

Baseada em conceitos fundamentais da física, química e biologia, a MD é uma técnica computacional que simula propriedades intrínsecas e extrínsecas de estruturas moleculares, além da dinâmica de sistemas físicos, químicos e biológicos mais complexos.[2] Desenvolvida a partir da década de 1950, com o uso emergente de computadores, a técnica tem sido aplicada em diversas áreas do conhecimento, destacando-se por diminuir o tempo de pesquisa em compostos candidatos, otimizar condições de síntese em reações e na manipulação de propriedades de materiais. Devido sua capacidade de prever propriedades e elucidar mecanismos em sistemas estudados, a MD permite investigar, por exemplo, reações químicas, propriedades termodinâmicas, estrutura molecular, propriedades físicas e químicas de materiais, difusão, transferência de calor, função de biomoléculas, entre outras[3]. Assim, tornou-se um recurso valioso para o estudo, pesquisa e desenvolvimento de materiais, biomoléculas e fármacos.[4]

O princípio da técnica consiste no desenvolvimento de um modelo matemático que descreva o comportamento e as interações das partículas no sistema simulado, no qual a precisão das simulações de MD está diretamente ligada aos modelos de força de campo utilizados.[5] Existem diversos tipos de modelos, como os de força de campo, potenciais empíricos, ab initio e híbridos, cada um voltado para diferentes aplicações.[6]

Os modelos são escolhidos durante a etapa de preparação do sistema nas simulações de MD, juntamente com a definição da estrutura inicial do sistema e das condições de contorno.[3] A escolha adequada do modelo de força de campo é crucial para garantir a precisão e a confiabilidade dos resultados obtidos nas simulações de MD, influenciando diretamente as trajetórias dos átomos ao longo do tempo e, consequentemente, as propriedades moleculares analisadas durante a etapa de análise dos resultados.[5] A MD tem a capacidade de fornecer informações detalhadas sobre a estrutura, dinâmica e propriedades em sistemas físicos, químicos e biológicos, permitindo a investigação de uma ampla gama de fenômenos em nível molecular.[6]

1.1 Aplicações da MD em Diversas Áreas

Na análise de mecanismos de reação, a MD possibilita desvendar esses mecanismos em nível molecular, acompanhando as etapas das reações e esclarecendo fatores que influenciam sua velocidade e seletividade.[2] No desenvolvimento de materiais catalíticos, a MD foi utilizada por Ramprasad *et al.* (2017) para desvendar o mecanismo de catálise homogênea em uma reação, detalhando a interação entre o catalisador e os reagentes.[3,7] A natureza dessa interação catalisador-reagentes é crucial para o desenvolvimento de novos catalisadores para processos químicos mais eficientes, abrindo caminho para a produção de produtos químicos mais limpos, eficientes e sustentáveis.[2]

Nas ciências biológicas e da saúde, a identificação da função de biomoléculas é facilitada pela capacidade da MD de visualizar a estrutura molecular em detalhes, revelando a posição e o arranjo dos átomos em uma molécula.[8] Esse tipo de informação é crucial, pois permite correlacionar a estrutura molecular, ou fragmentos desta, às propriedades físico-químicas e à atividade biológica da biomolécula. Estudos anteriores relatam o uso MD para elucidar a estrutura tridimensional de uma proteína envolvida em uma doença genética, a fim de estudar o desenvolvimento de novos possíveis tratamentos.[9] A abordagem integrada de diversos parâmetros de forma detalhada proporcionada pela MD representa um avanço significativo no campo da biologia molecular e da medicina.

A DM também é aplicada na predição de propriedades termodinâmicas, como energia livre de Gibbs, entalpia, entropia e capacidade calorífica em diferentes condições de temperatura e

pressão.[5] Estudos anteriores mostram que a MD é uma poderosa maneira de se calcular a entropia de um material polimérico, proporcionando visões valiosas sobre sua estabilidade térmica e comportamento em diferentes condições ambientais.[9]

A ciência de materiais se beneficia do uso da MD, pois as simulações permitem prever propriedades físico-químicas, como condutividade elétrica, condutividade térmica, magnetismo e resistência mecânica de novos materiais. [7,10] Tais simulações são utilizadas para buscar a otimização das propriedades dos materiais para atender à aplicação desejada.[11] Estudos publicados sobre o uso da MD no *design* de novos materiais são recorrentes, como o trabalho de Smith J. (2023), no qual a MD foi utilizada no projeto de um material com alta condutividade térmica, visando aumentar a eficiência de dispositivos eletrônicos,[12] bem como o estudo de Pilania, G., *et al.* (2021), que desenvolveu um catalisador para a conversão de gás carbônico em metanol por hidrogenação.[11] Além disso, a MD possibilita o *design* de materiais funcionais com aplicações promissoras em áreas como energia solar, armazenamento de energia e eletrônica molecular.[13,14]

1.2 Desafios e perspectivas futuras da MD

Embora a MD ofereça significativos benefícios em termos de economia de tempo e recursos na pesquisa e desenvolvimento, ela possui algumas barreiras de entrada que, atualmente, podem ser grandes obstáculos para sua aplicação de forma mais ampla. Uma das principais barreiras é o custo computacional, tanto em termos de capacidade de processamento de dados do hardware, quanto ao custo financeiro relacionado. Dependendo da complexidade dos modelos utilizados e do tamanho do sistema simulado pelo algoritmo, o uso da MD pode se tornar inviável cronológica e economicamente.[15,16,17]

Outro ponto crítico é a escolha adequada ou o desenvolvimento de um modelo matemático otimizado para a aplicação pretendida. Essa escolha é fundamental para garantir a qualidade e a precisão das simulações, e o desenvolvimento de novos modelos e métodos de simulação mais eficientes é uma área ativa de pesquisa em MD.[1,5,9] Além disso, a interpretação dos dados obtidos por simulação requer expertise técnica e a integração com outras técnicas experimentais, tornando o viés humano um aspecto importante a ser considerado durante o desenvolvimento do projeto.[18]

Ainda assim, a MD é uma ferramenta com um futuro promissor, especialmente devido aos avanços na computação de alto desempenho e no desenvolvimento de novos métodos de simulação. Esse progresso amplia a capacidade de estudo de sistemas cada vez mais realistas e complexos, possibilitando uma maior compreensão dos fenômenos em microescala e suas relações com as propriedades macroscópicas moleculares. Nesse sentido, a MD tem potencial de transformar diversos setores, como a indústria farmacêutica, energia, materiais inorgânicos e biotecnológicos, oferecendo novas formas para o estudo de fenômenos em sistemas físicos, químicos e biológicos.[9]

1.3 O advento da *Artificial Intelligence* (AI) e do *Machine Learning* (ML) na computação

A concepção da AI tem suas raízes na filosofia, matemática e ciência da computação do início do século XX, com o objetivo de criar agentes inteligentes capazes de raciocinar, aprender e agir de forma autônoma.[7] Um marco importante na história da AI foi o teste de Turing, proposto por Alan Turing em 1950, que visava avaliar a capacidade de uma máquina de mimetizar o comportamento inteligente equivalente ao de um humano.[19] Nas últimas décadas, o campo da AI vivenciou um crescimento exponencial, impulsionado pelo desenvolvimento de novos algoritmos, pelo aumento da disponibilidade de dados e pela elevação da capacidade de processamento computacional

Tradicionalmente, os algoritmos de AI eram configurados com base em sistemas de regras e lógica explicitamente programados no código-fonte do algoritmo. Como alternativa, o ML surgiu no início do século XX, fundamentado na teoria estatística e na otimização matemática, permitindo que algoritmos se ajustem automaticamente a partir de dados de treinamento, sem a necessidade de programação explícita para cada tarefa.[20] Um exemplo icônico de ML são as Redes Neurais Artificiais, que são inspiradas no funcionamento do cérebro humano, simulando uma rede de neurônios artificiais interconectados que processa os dados matematicamente de acordo com uma função programada, para identificar padrões em conjuntos de dados e fazer previsões sobre dados futuros. O ML tem suas raízes na teoria estatística e na otimização matemática do início do século XX.[20] Com o avanço da tecnologia e o aumento da disponibilidade de grandes volumes de dados, o ML tornou-se uma ferramenta essencial em diversas áreas, incluindo reconhecimento de padrões, processamento de linguagem natural e análise preditiva.[21,22]

2. Técnicas de ML e Tipos de AI

As técnicas de ML mais comumente utilizadas são classificadas em três tipos: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço.[15,16,21] Cada uma dessas técnicas adota uma abordagem diferente em relação ao tratamento dos dados de entrada pelo algoritmo de AI.

No aprendizado supervisionado, o modelo de AI utiliza uma amostragem de dados de entrada (*input*) e saída (*output*) já rotulados, ou seja, os dados de entrada e suas saídas correspondentes são previamente conhecidos. A partir desses "dados de treinamento", o modelo realiza previsões sobre novos dados que não estão no conjunto de treino.[13,15,16,21]

No aprendizado não supervisionado, o algoritmo parte de uma base de dados sem rótulos, ou seja, não há informações prévias de correspondência de entrada e saída. O objetivo do algoritmo é identificar padrões ou estruturas nos dados não rotulados e fazer previsões apropriadas.[13,15,16,21] Também é possível que o algoritmo parta de uma mescla de dados rotulados e não rotulados para criar os dados de treino, sendo esta abordagem classificada como aprendizado semi-supervisionado. As vantagens principais em fazê-lo são para tornar o modelo mais robusto mediante ao aumento da quantidade de dados disponíveis e reduzir de custo de aquisição de dados, uma vez que dados não rotulados geralmente são mais baratos.[17]

Por fim, no aprendizado por reforço, o método de aprendizagem é baseado em um sistema de recompensa e punição. O algoritmo interage com o ambiente e aprende quais ações maximizam sua recompensa acumulada. A precisão das predições do algoritmo é ajustada de acordo com a recompensa recebida, e os ajustes que aumentam a precisão são favorecidos. Esta abordagem é comumente utilizada em sistemas onde a sequência de decisões é importante, como em robótica e jogos.[15]

No contexto do ML, o tipo de algoritmo é a parte essencial das aplicações em AI, pois a modelagem e estrutura do algoritmo utilizado influenciam diretamente as propriedades, vantagens e limitações do modelo a ser treinado.Uma possível classificação dos algoritmos é pela finalidade: classificação ou regressão.[21] Embora os algoritmos possam ser usados para finalidades diferentes, sua

performance varia de acordo com a aplicação devido às suas propriedades matemáticas.Os algoritmos de classificação têm como objetivo diferenciar dados e agrupá-los de acordo com a semelhança dentro de um conjunto de dados.[16] Nesse contexto, esses algoritmos são eficientes para rotular dados conforme classes definidas. Por exemplo, para reconhecimento de idioma em um texto, o *output* do algoritmo será o idioma identificado; mesmo que existam fragmentos de texto em outro idioma, a escrita predominante guiará a classificação. Por outro lado, os algoritmos de regressão visam prever valores contínuos a partir de dados de entrada, da mesma forma que uma equação estabelece a relação entre valores de entrada e saída.[16]

Há uma variedade de algoritmos com diferentes características e aplicações que podem ser usados em ML, muitos dos quais são amplamente aplicados no campo da química.[23, 24] Alguns exemplos incluem:

***k-Nearest Neighbors* (kNN)**

Este modelo baseia-se na comparação de um *input* com uma base de dados rotulada. O algoritmo classifica o *input* de acordo com os "k" dados rotulados mais semelhantes (onde "k" representa o número de vizinhos considerados na classificação).[16,17,21] A figura abaixo demonstra o processo de classificação do algoritmo kNN.

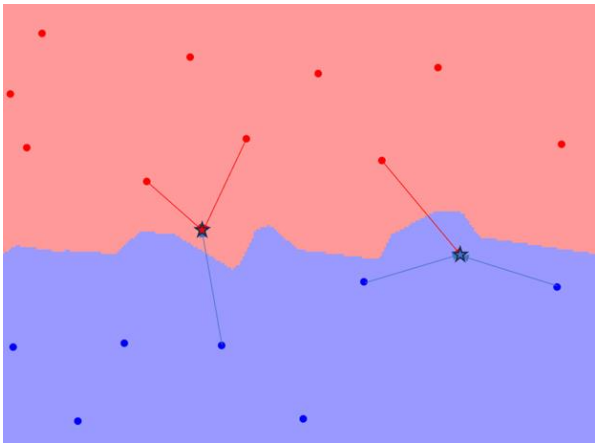

Figura 1: Predição feita por um algoritmo kNN (k=3) em uma base de dados genérica.

O algoritmo kNN é considerado um modelo simples, possuindo dois principais parâmetros: o número "k" de vizinhos e a metodologia para medir as distâncias entre os dados.[21] A simplicidade do modelo oferece vantagens como fácil compreensão das predições, flexibilidade para ajustes a fim de fornecer uma predição aceitável e uma rápida execução utilizando um conjunto de dados de treinamento. Esses fatores fazem com que o kNN seja atrativo para aplicações onde, para os desenvolvedores, a interpretabilidade do modelo é importante.[21]

Mesmo que muito usado como modelo exploratório inicial o kNN é preterido devido a algumas limitações práticas. O algoritmo requer uma base de dados rotulada e, para cada predição, realiza uma busca por similaridade em toda a base de dados, resultando em um alto custo computacional, especialmente para conjuntos de dados grandes.[15] Além disso, modelos não paramétricos como o kNN apresentam dificuldades para lidar com diferentes formas de distribuição dos dados, particularmente em situações que envolvem múltiplos *inputs* ou conjuntos de dados esparsos.[15,21]

***Support Vector Machine/Regression* (SVM/SVR)**

O modelo de SVM baseia-se no conceito de hiperplano aplicado a um conjunto de dados. Um hiperplano é uma superfície de dimensionalidade inferior que representa uma "fronteira" que separa diferentes classes de dados, conforme exemplificado na figura a seguir.[16] O SVM visa encontrar o hiperplano que maximize a margem de separação entre as classes, sendo amplamente utilizado em problemas de classificação binária. Já o modelo de SVR é uma extensão do SVM, voltada para a regressão, buscando prever valores contínuos com base nas características de entrada.

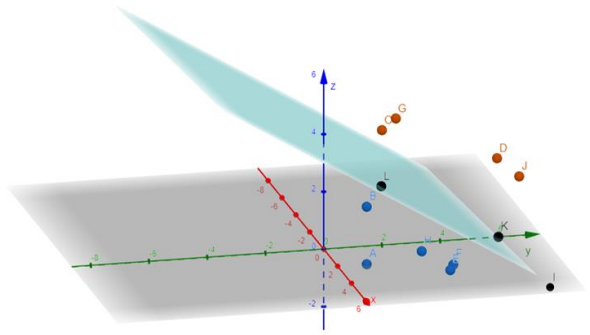

Figura 2: Representação de um hiperplano em um plano 3D.

Uma das qualidades intrínsecas do SVM é a eficácia em conjuntos de dados com alta dimensionalidade. O modelo é robusto, mesmo quando o número de dimensões é maior do que o número de amostras.[13] Em contrapartida, é necessário realizar um pré-processamento cuidadoso dos dados de treinamento, de forma que os hiperparâmetros do algoritmo possam proporcionar uma classificação adequada. Embora o SVM mantenha sua eficácia com um volume reduzido de dados, ele não escala bem em termos de demanda computacional, o que pode ser um fator limitante.[13,15]

***Decision Tree* (DT) & *Random Forest* (RF)**

A DT é um algoritmo que se baseia no sequenciamento de nós com estruturas de decisão do tipo "*if-else*" (se verdadeiro, x; se falso, y).[17,21] A RF é uma combinação de múltiplas DT's para melhorar a precisão e reduzir o risco de *overfitting*, utilizando um processo de *bootstrap* e agregação para gerar e combinar diferentes DT's, o que proporciona maior robustez ao modelo. Esse método de tomada de decisão é intuitivo e de fácil compreensão, sendo muitas vezes representado em diagramas de fluxo, como mostrado na figura 3.

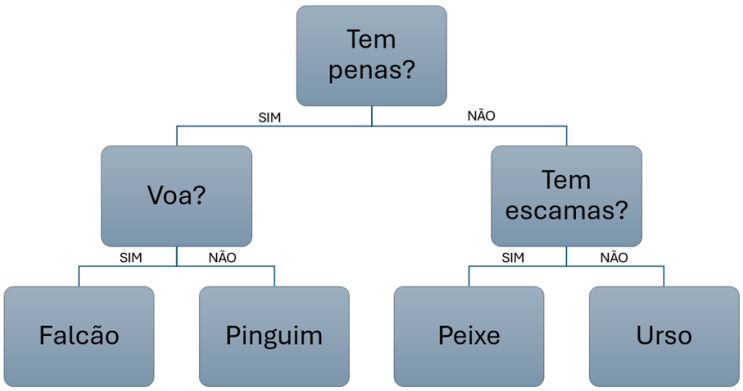

Figura 3: DT para classificação de um *input* em animais

O algoritmo de DT é amplamente utilizado em problemas de classificação, embora também possa ser empregado em regressão. Ele possui diversas vantagens no seu uso, apesar de apresentar algumas desvantagens críticas em termos de predição.[13,21]

Segundo Bishop (2006)[15], as principais qualidades das DT's são: facilidade de compreensão, robustez a dados anômalos, capacidade de lidar com variáveis contínuas e discretas, seleção automática de variáveis, redução da dimensionalidade e do *overfitting*, além de serem relativamente fáceis de treinar e capazes de lidar com grandes volumes de dados.[15]

No entanto, as DT's apresentam um problema de instabilidade, em que pequenas mudanças nos dados de treinamento podem ser amplificadas nas predições, devido ao caráter hierárquico do fluxo de dados. Esse problema ocorre porque um erro em uma etapa da DT se propaga para o restante da árvore. Para mitigar tal efeito, algumas técnicas são frequentemente utilizadas:

- ***Bagging/Bootstrap***: Técnica em que múltiplas DT's independentes são treinadas com amostras aleatórias do conjunto de dados inicial, resultando em variações sutis nas predições feitas pelo conjunto de DT's.[15]
- ***Random Forests***: Aumentam o grau de aleatoriedade no processo de *Bagging*, atribuindo aleatoriedade adicional aos subconjuntos associados aos nós de uma DT.[17]

- ***Boosting***: Consiste em uma sequência de DT's, na qual cada modelo subsequente tenta corrigir os erros do modelo anterior, ajustando-se com base nas predições anteriores.[15]

O uso dessas técnicas tem como objetivo inserir maior aleatoriedade no processo, reduzir o efeito de amplificação de erros, diminuir o *overfitting* e aumentar a robustez dos modelos.[15,21]

***Logistic Regression* (LR)**

O algoritmo de LR é um modelo probabilístico utilizado para classificação, que prevê a probabilidade de um *input* pertencer a uma classe entre "n" possíveis classes.[1] Embora seja classificado como um algoritmo de regressão linear, ele é frequentemente mais eficaz em problemas de classificação.[15,21]

Os pontos positivos do modelo incluem a facilidade de treinamento e de execução das predições, boa escalabilidade para grandes volumes de dados, e o fornecimento de informações sobre como a predição foi realizada. Contudo, em muitas situações, é difícil entender a relação dos coeficientes com os inputs fornecidos, o que dificulta a compreensão do modelo.[17,21]

***k-Means Clustering* (kMC)**

O algoritmo kMC é um método não paramétrico utilizado para agrupar dados de uma distribuição em diferentes grupos (clusters). Seu princípio de funcionamento baseia-se na partição dos dados, identificando agrupamentos de acordo com a proximidade dos pontos.[15,17,21]

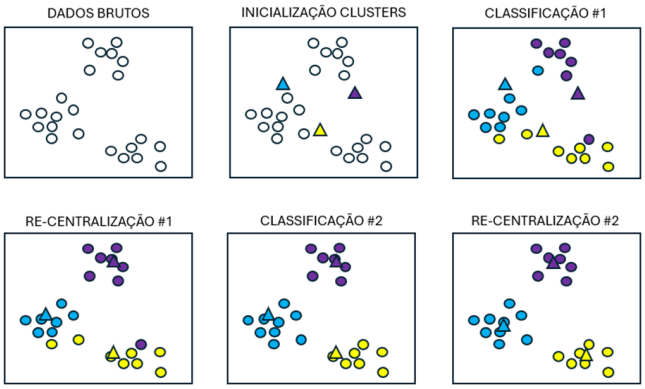

Figura 4: Processo de partição e caracterização dos agrupamentos em um kMC

O modo de trabalho do algoritmo kMC é semelhante ao algoritmo kNN, mas com a introdução de um novo elemento: os *clusters*. O algoritmo segue um ciclo de procedimentos, começando com a atribuição de um dado ao *cluster* mais próximo, calculando a distância do ponto de dado ao centro do *cluster*. Em seguida, a posição do centro do *cluster* é recalculada com base nos novos membros atribuídos ao *cluster*.[15,21] Este processo é repetido até que todos os dados de *input* tenham sido processados pelo algoritmo e os centros dos *clusters* se estabilizem.

Assim como o kNN, o kMC é um algoritmo relativamente simples e rápido de implementar e treinar. No entanto, apresenta limitações específicas em relação aos dados que pode processar e às características intrínsecas do próprio algoritmo.[21] Inicialmente, deve-se declarar ao algoritmo o número de *clusters* desejados, o que implica que é necessário ter algum conhecimento prévio sobre as características da amostragem para determinar essa quantidade adequadamente.[15,21] Outro aspecto importante é que o algoritmo assume que os *clusters* possuem formas simétricas e tamanhos equivalentes, o que o torna eficaz para dados próximos uns dos outros, mas limita sua performance quando os dados são dispersos ou distribuídos de maneira não uniforme.[17]

***Principal Components Analysis* (PCA)**

O algoritmo PCA é um método de regressão não supervisionado, cuja principal característica é a redução da dimensionalidade dos dados, ajustando a influência de diferentes variáveis com base nas características mais relevantes para a análise. Esse ajuste é realizado através da transformação e manipulação dos dados de modo a identificar as "componentes principais", ou seja, os eixos que explicam a maior parte da variabilidade presente no conjunto de dados.[15,21]

O PCA é especialmente útil quando se deseja simplificar um conjunto de dados de alta dimensionalidade, mantendo a maior parte da informação relevante. Através da identificação dos componentes principais, o algoritmo consegue diminuir o número de variáveis do modelo, facilitando a visualização e a análise, e permitindo uma melhor compreensão das relações entre as variáveis originais.[21]

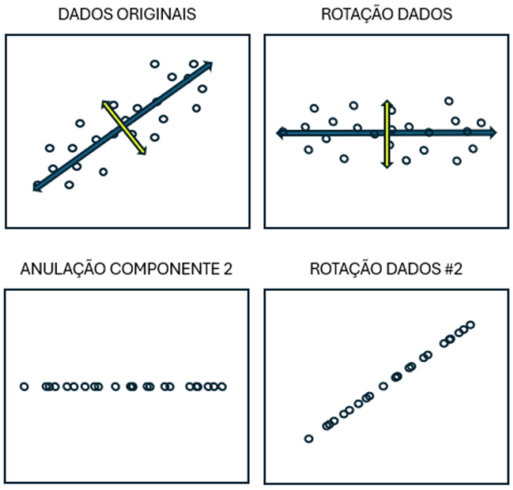

Figura 5: Plotagem gráfica de uma regressão PCA genérica.

Conforme mostrado na figura 5, através da transformação matemática dos dados realizada por PCA, é possível identificar a influência de cada uma das características analisadas no gráfico de distribuição. Essa análise permite visualizar as principais variáveis que contribuem para a variabilidade dos dados, facilitando a compreensão dos padrões e estruturas presentes no conjunto de dados.

***Artificial Neural Network* (ANN) e *Deep Learning* (DL)**

O modelo de ANN é inspirado na forma como os neurônios do cérebro humano interagem entre si para processar informações. Uma ANN consiste em uma série de camadas que possuem funções específicas e são responsáveis pelo processamento dos dados de entrada até a obtenção dos resultados finais.[16]

- ***Input Layer***: Nesta camada, são inseridos os dados de entrada no algoritmo. Pode haver uma etapa de pré-processamento dos dados, dependendo do tipo de aplicação e das características dos dados.

- ***Hidden Layer***: Esta camada contém uma rede de neurônios artificiais interligados que recebem os dados da camada de entrada (*Input Layer*). Esses neurônios aplicam funções de ativação e utilizam hiperparâmetros para transformar e processar as informações recebidas.

- ***Output Layer***: Camada onde os dados de saída são gerados após o processamento pela rede. Pode haver um processo de atualização dos parâmetros da função de ativação com base na precisão dos dados de saída, o que ajuda a otimizar a performance do modelo.

O fluxo de dados em uma ANN segue um caminho hierárquico, onde cada camada transforma as informações recebidas e transmite os resultados para a camada seguinte, até que a saída seja obtida na *Output Layer*. Esse processo é representado pela figura a seguir, que ilustra o fluxo de dados através das camadas da rede.

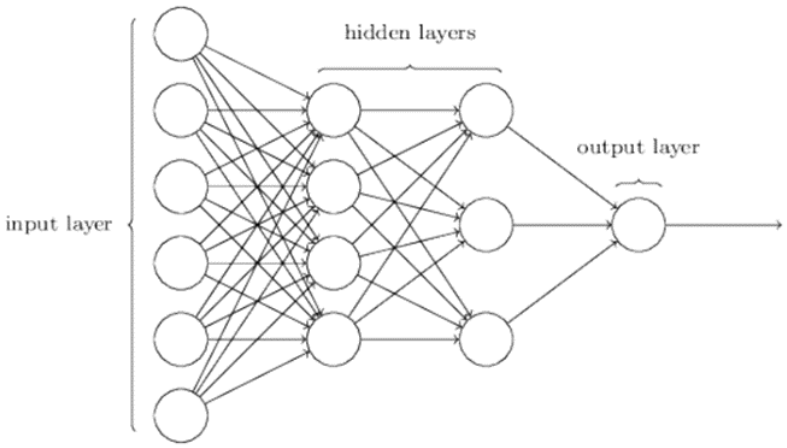

Figura 6: Arquitetura e fluxo de dados numa ANN.[16]

O "cérebro" da ANN está na função de ativação matemática aplicada aos dados de entrada. Essas funções introduzem não-linearidades nos neurônios artificiais, permitindo que a rede modele as relações complexas entre os dados de entrada e saída. A ausência de uma função de ativação não-linear limitaria a capacidade de generalização da ANN, restringindo o modelo a relações lineares simples.[21]

As funções de ativação desempenham um papel crucial, pois determinam como os neurônios da rede processam as informações recebidas e afetam diretamente a capacidade do modelo de aprender padrões complexos nos dados. Algumas das funções de ativação mais conhecidas são apresentadas na tabela 1 e suas representações gráficas na figura 7:

Tabela 1: Funções de ativações em ANN's.

Função de Ativação	Aplicação
Degrau (Perceptrons)	Classificação
Sigmoidal	Classificação
Tangente Hiperbólica (tanh)	Classificação
Base Radial (RBF)	Classificação
Limiar Linear	Regressão
Unidade Linear Retificada (ReLU)	Regressão
Gaussian Error Linear Unit (GELU)	Regressão

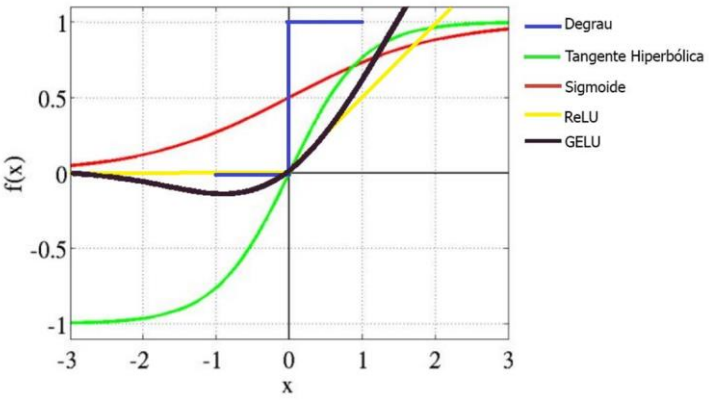

Figura 7: Plotagem gráfica das funções de ativação.

A função de ativação ReLU (Rectified Linear Unit) é uma das mais utilizadas, especialmente em redes profundas, devido à sua simplicidade e eficácia na mitigação do problema do gradiente desaparecente em redes neurais. A ReLU transforma os valores negativos em zero e mantém os valores positivos, o que facilita a convergência durante o treinamento. Funções como a sigmoidal e a tangente hiperbólica são também amplamente usadas em redes mais rasas para problemas de classificação, enquanto funções como GELU são mais comuns em arquiteturas complexas, como aquelas usadas em modelos de Deep Learning.[21]

As ANN's têm como principais pontos fortes a capacidade de processar grandes volumes de dados, criando modelos preditivos de alta complexidade. No entanto, para que esta ferramenta produza resultados adequados, ela depende fortemente de três fatores principais: a capacidade de processamento computacional, uma base de dados de treinamento ampla e a configuração correta dos hiperparâmetros das funções de ativação.[16]

No primeiro ponto, quanto maior for o grau de complexidade de uma ANN, determinado pelo número de neurônios e camadas de neurônios, maior será a demanda computacional para processar as operações matemáticas de cada função de ativação em cada neurônio, impactando significativamente o tempo de treinamento. Isso se torna um desafio particularmente relevante em problemas complexos, onde muitas camadas ocultas são necessárias para obter bons resultados.

Em relação ao segundo ponto, é importante ressaltar que, embora as ANN's possam ser utilizadas em diferentes tipos de aprendizado, supervisionados ou não, elas tendem a performar melhor quando treinadas com dados relativamente homogêneos e bem estruturados. Isso significa que uma base de dados de alta qualidade pode ter um impacto direto na eficiência e precisão do modelo.

O terceiro ponto diz respeito à sensibilidade aos hiperparâmetros das funções de ativação, que são fundamentais para a eficiência do modelo preditivo em execução. A escolha dos hiperparâmetros adequados pode ser realizada com o auxílio de funções de erro, ajustando os pesos da rede de acordo com as predições realizadas anteriormente. Esse processo, no entanto, requer intervenção humana e configura uma etapa crítica no desenvolvimento do modelo.[17]

Com o avanço da tecnologia e o aumento significativo da capacidade de processamento de dados dos computadores, as Redes Neurais puderam crescer em quantidade de nós e na complexidade das conexões nas camadas ocultas (*Hidden Layers*), permitindo a criação de redes cada vez mais profundas.[14] Esses modelos de aprendizagem são chamados de *Deep Learning*, pois são compostos por "várias camadas ocultas, arquiteturas de conectividade complexas e diferentes operadores de transferência"[23]. Esse avanço permitiu que as redes neurais profundas fossem aplicadas a problemas extremamente complexos, como reconhecimento de imagens e processamento de linguagem natural, que antes eram considerados inviáveis.[14]

3. Integração de ML e MD

O desenvolvimento de um modelo preditivo por aprendizado de máquina (ML) segue um processo que envolve várias etapas fundamentais, tais como: preparação dos dados, seleção do algoritmo, treinamento, validação, otimização e aplicação. A preparação dos dados utilizados para o treinamento de um modelo de ML, desde que não se trate de aprendizado não supervisionado, envolve a formatação consistente dos dados (incluindo unidades de medida e rótulos) e a associação entre dados de entrada e saída esperados, criando pares de entrada-saída. No contexto da Dinâmica Molecular (MD), os dados de entrada geralmente são as coordenadas atômicas de um sistema molecular obtidas por meio de simulações de MD ou experimentos, enquanto os dados de saída correspondem às propriedades moleculares de interesse. O uso de técnicas de ML em ciência dos materiais tem se

mostrado promissor devido à sua capacidade de prever propriedades moleculares a partir de grandes quantidades de dados experimentais ou simulados, como discutido por Schmidt et al. (2019).[12]

Uma vez estabelecido um conjunto de dados, é necessário selecionar qual algoritmo de ML é mais apropriado para a tarefa em questão, seja ela classificação ou regressão. O modelo é treinado utilizando uma fração dos dados disponíveis, ajustando os parâmetros do algoritmo de acordo com os dados de treinamento fornecidos. Métodos como a validação cruzada podem ser empregados para evitar o problema do *overfitting* e garantir a capacidade de generalização do modelo.[25]

O desempenho do modelo de ML é avaliado utilizando um conjunto de dados de validação, não utilizados durante o treinamento. As respostas do modelo para esses dados de validação, cujas respostas corretas são conhecidas, são avaliadas por meio de métricas estatísticas, como desvio padrão e erro médio absoluto, entre outros parâmetros, para validação e verificação da precisão do modelo. A otimização do modelo pode ser realizada ajustando seus hiperparâmetros, que são os parâmetros que controlam o aprendizado do modelo. Técnicas como a busca em grade (grid search) ou otimização bayesiana podem ser utilizadas para encontrar a melhor configuração dos hiperparâmetros e, assim, melhorar a precisão das previsões.[26]

Uma vez treinado e validado, o modelo de ML pode ser utilizado para prever propriedades moleculares em novos sistemas que não foram incluídos no conjunto de dados de treinamento, sendo aplicável aos objetivos propostos para o desenvolvimento do algoritmo preditivo.

3.1 A Sinergia entre ML e MD

O uso de ML na MD oferece diversas vantagens significativas. Uma delas é a redução do tempo computacional, pois os modelos de ML são capazes de prever propriedades moleculares com alta precisão em um tempo muito menor do que as simulações de MD completas, possibilitando o estudo de sistemas moleculares maiores e mais complexos.[22] Além disso, o uso de algoritmos de ML pode aprimorar a precisão das previsões, ao capturar relações complexas entre variáveis moleculares que métodos tradicionais baseados em força bruta podem não conseguir identificar. Ademais, o ML pode ser empregado em diversas etapas do processo de MD como na otimização de parâmetros de simulação, substituindo modelos de interações interatômicas, otimização de amostragem durante as

simulações e pós-processamento dos dados obtidos com as simulações.[27] Outra vantagem é a exploração de um amplo espaço químico, onde modelos de ML são capazes de prever propriedades de compostos virtuais não sintetizados, facilitando a descoberta de novos materiais e fármacos em um ritmo mais rápido.[22] Além disso, os modelos de ML podem ajudar a identificar padrões e relações ocultas nos dados gerados pelas simulações de MD, fornecendo insights valiosos sobre os mecanismos moleculares por trás de processos químicos e biológicos. A interpretabilidade dos modelos de ML também pode ser um fator importante para a extração de conhecimento científico a partir das previsões realizadas.

3.2 Desafios do Uso de ML em MD

Embora a aplicação de ML na MD ofereça muitas vantagens e possa superar diversas barreiras de entrada associadas ao uso de técnicas de MD, existem dificuldades inerentes ao emprego conjunto dessas abordagens. Os modelos de ML mais robustos, utilizados em aplicações de MD, requerem grandes quantidades de dados de treinamento de alta qualidade para aprender de forma eficaz.[9] A obtenção de dados experimentais de propriedades moleculares é muitas vezes cara e demorada e as simulações de MD para gerar esses dados também podem ser computacionalmente dispendiosas. Técnicas como o aprendizado por transferência, que adapta modelos treinados em uma tarefa para resolver uma tarefa relacionada, são estratégias utilizadas para lidar com a limitação de dados.

Além da questão da qualidade e quantidade de dados de treinamento, a falta de interpretabilidade dos modelos de ML, especialmente das redes neurais artificiais complexas, pode dificultar a compreensão dos mecanismos de predição e limitar a confiança nos resultados obtidos.[9] Técnicas como o LIME (Local Interpretable Model-Agnostic Explanations) podem ser aplicadas para explicar as previsões de modelos complexos e melhorar a interpretabilidade dos resultados.

Na seleção dos conjuntos de dados para treinamento, é essencial garantir uma amostragem representativa para evitar problemas de *underfitting* e *overfitting*. O *underfitting* ocorre quando um modelo de ML não é suficientemente complexo para capturar os padrões dos dados, enquanto o *overfitting* ocorre quando o modelo se ajusta excessivamente aos dados de treinamento, prejudicando sua capacidade de generalizar para novos dados.[9] O *overfitting* pode resultar em previsões imprecisas

para moléculas que não estão no domínio dos dados de treinamento. Para mitigar o *overfitting*, técnicas como validação cruzada e regularização são amplamente empregadas.

Além disso, é necessário atentar-se ao caráter antropológico no processo de desenvolvimento de um modelo de ML, pois as decisões relacionadas à escolha dos algoritmos, dos dados de treinamento e dos ajustes de parâmetros são feitas pelo programador responsável pelo modelo. Essas escolhas introduzem um viés humano significativo, que pode impactar diretamente a qualidade das predições realizadas pelo algoritmo. Portanto, é fundamental que o desenvolvimento do modelo considere a mitigação desse viés, para garantir a robustez e a confiabilidade dos resultados.

CONCLUSÃO

A integração entre ML e MD representa uma evolução significativa na predição de propriedades moleculares, contribuindo para avanços em diversas áreas da ciência e engenharia. Ao longo deste trabalho, discutimos como algoritmos de ML, incluindo kNN, kMC, SVM, RF, ANN, RL e PCA, podem ser utilizados para otimizar processos em MD, reduzindo significativamente a demanda computacional e aumentando a precisão das previsões de propriedades moleculares. Essa sinergia possibilita a análise de sistemas moleculares maiores e mais complexos o que seria inviável com métodos tradicionais de MD ampliando o potencial de descoberta de novos materiais e fármacos.

Apesar dos avanços, o uso combinado de ML e MD enfrenta desafios importantes, como a necessidade de grandes volumes de dados de alta qualidade, os altos custos computacionais associados às simulações e a obtenção de dados experimentais, além da dificuldade de interpretar modelos complexos, especialmente aqueles baseados em redes neurais profundas. Esses obstáculos podem ser mitigados por estratégias como a aprendizagem por transferência, técnicas de regularização e o uso de métodos como LIME para melhorar a interpretabilidade dos modelos. Ainda assim, o desenvolvimento de novas técnicas de simulação, maior capacidade de processamento e aprimoramento dos algoritmos de ML prometem superar esses desafios no futuro.

Olhando para o futuro, o avanço da computação de alto desempenho, aliado ao desenvolvimento de métodos mais eficientes de ML, pode expandir ainda mais a aplicabilidade da MD, permitindo uma exploração mais profunda dos fenômenos moleculares e facilitando a criação de soluções inovadoras para desafios nas indústrias farmacêutica, de materiais e de energia. Assim, a sinergia entre ML e MD continua a ser um campo promissor, com potencial para impactar positivamente a ciência e a tecnologia, promovendo um maior entendimento dos sistemas moleculares e possibilitando novas descobertas.

REFERÊNCIAS

1. FRENKEL, D.; SMIT, B. *Understanding Molecular Simulation: From Algorithms to Applications*. San Diego: Academic Press, 2002.

2. MARK, P.; JENSEN, L. *Physical Modeling of Biological Systems: Principles of Simulations*. Springer, 2000.

3. FERNANDES, F. M. S.; FARTARIA, R. P. S. Gibbs ensemble Monte Carlo: Algorithm and implementation for gas-liquid coexistence. *American Journal of Physics*, 2015. doi: https://doi.org/10.1119/1.4921392.

4. ANDRIJAUSKAS, F.; CATROLI, G. F. Computação de alto desempenho e dinâmica molecular. In: *High Performance Computing and Molecular Dynamics*, 2020. doi: https://doi.org/10.22533/at.ed.4672028093

5. WARSHEL, A. *Computer Modeling of Chemical Reactions in Enzymes and Solutions*. New York: John Wiley & Sons, 2006.

6. ALLEN, M. P.; TILDESLEY, D. J. *Computer Simulation of Liquids*. 1ª ed. New York: Oxford University Press, 1987.

7. RAMPRASAD, R.; BATRA, R.; PILANIA, G.; MANNODI-KANAKKITHODI, A.; KIM, C. Machine learning in materials informatics: recent applications and prospects. *npj Computational Materials*, v. 3, 2017. doi: https://doi.org/10.1038/s41524-017-0056-5.

8. HIMANEN, L.; JÄGER, M. O. J.; MOROOKA, E. V.; CANOVA, F. F.; RANAWAT, Y. S.; GAO, D. Z.; RINKE, P.; FOSTER, A. S. DScribe: Library of Descriptors for Machine Learning in Materials Science. *Computational Physics Communications*, v. 247, p. 106949, 2020. doi: https://doi.org/10.1016/j.cpc.2019.106949.

9. WEI, J.; CHU, X.; SUN, X.-Y.; et al. Machine learning in materials science. *InfoMat*, v. 1, n. 3, p. 338-358, 2019. doi: https://doi.org/10.1002/inf2.12028.

10. ALLEN, M. P.; TILDESLEY, D. J. *Simulação de Dinâmica Molecular: Teoria e Aplicação*, 2017.

11. PILANIA, G. Machine learning in materials science: From explainable predictions to autonomous design. *Computational Materials Science*, v. 193, p. 110360, 2021. doi: https://doi.org/10.1016/j.commatsci.2021.110360.

12. SMITH, John. Effects of Machine Learning Algorithms for Predicting and Optimizing the Properties of New Materials in the United States. *European Journal of Physical Sciences*, v. 6, n. 1, p. 23-34, maio 2023. doi: https://doi.org/10.47672/ejps.1444.

13. GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge: MIT Press, 2016.

14. LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, p. 436-444, 2015. doi: https://doi.org/10.1038/nature14539.

15. BISHOP, C. M. *Pattern Recognition and Machine Learning*. 1ª ed. New York: Springer, 2006. ISBN 978-0387310732. doi: https://doi.org/10.1007/978-0-387-45528-0.

16. NIELSEN, M. A. *Neural Networks and Deep Learning*. San Francisco: Determination Press, 2015.

17. MURPHY, K. P. *Probabilistic Machine Learning: An Introduction*. Cambridge: MIT Press, 2022.

18. RAPAPORT, D. C. *The Art of Molecular Simulation*. Cambridge: Cambridge University Press, 2004.

19. TURING, A. M. Computing machinery and intelligence. *Mind*, London, 1950.

20. MITCHELL, T. M. *Machine Learning*. New York: McGraw-Hill, 1997.

21. MULLER, A. C.; GUIDO, S. *Introduction to Machine Learning with Python*. 1ª ed. Schanafelt, D. (ed.). Sebastopol: O'Reilly Media, 2017.

22. SCHMIDT, J.; MARQUES, M. R. G.; BOCHEM, J.; BÜCKER, H. M.; SCALABRIN, A. Recent advances and applications of machine learning in materials science. *npj Computational Materials*, v. 5, p. 83, 2019. doi: https://doi.org/10.1038/s41524-019-0221-0.

23. TINA G. M.; VENTURA C.; FERLITO S.; DE VITO S.; Applied Sciences, 2021, v. 11, n. 16, p. 7550, doi: https://doi.org/10.3390/app11167550.

24. TRIHN, C.; MEIMAROGLOU, D.; HOPPE, S.; Processes, 2021, v. 9, n. 8, doi: https://doi.org/10.3390/pr9081456.

25. KOHAVI, R.; A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Em: Proceedings of the second international conference on knowledge discovery and data mining, Montreal. Canadá. 1995. p 188-192.

26. SNOEK, J.; LAROCHELLE, H.; ADAMS, R. P.; Advances in Neural Information Processing Systems. Em Annual Conference on Neural Information Processing Systems. Lake Tahoe. United States. 2012. v.4. p 2951-2959.

27. SHUZHE, W., SEREINA, R. Machine Learning in the Area of Molecular Dynamics Simulations. Royal Society of Chemistry. 2020. doi: https://doi.org/10.1039/9781788016841-00184