UNIVERSIDADE FEDERAL DE PELOTAS
Instituto de Física e Matemática
Programa de Pós-Graduação em Física

Dissertação de Mestrado

Aprendizado de máquina aplicado para classificação de fases em sistemas de matéria mole

Vinicius Fonseca Hernandes

Pelotas, 2021

Vinicius Fonseca Hernandes

Aprendizado de máquina aplicado para classificação de fases em sistemas de matéria mole

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Física do Instituto de Física e Matemática da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Mestre em Física.

Orientador: Prof. Dr. José Rafael Bordin
Coorientador: Prof. Dr. Mario Lucio Moreira

Pelotas, 2021

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

H557a Hernandes, Vinicius Fonseca

Aprendizado de máquina aplicado para classificação de fases em sistemas de matéria mole / Vinicius Fonseca Hernandes ; José Rafael Bordin, orientador ; Mário Lúcio Moreira, coorientador. — Pelotas, 2021.

101 f. : il.

Dissertação (Mestrado) — Programa de Pós-Graduação em Física, Instituto de Física e Matemática, Universidade Federal de Pelotas, 2021.

1. Redes neurais. 2. Dinâmica molecular. 3. Soluções aquosas. I. Bordin, José Rafael, orient. II. Moreira, Mário Lúcio, coorient. III. Título.

CDD : 006.3

Elaborada por Ubirajara Buddin Cruz CRB: 10/901

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE PELOTAS
PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
COORDENAÇÃO DE PÓS-GRADUAÇÃO

DEFESA DE DISSERTAÇÃO

Aluno	20103171 - VINICIUS FONSECA HERNANDES		
CPF	03770707001	**Nacionalidade**	BRASILEIRA
Naturalidade	PELOTAS		
Ingresso	SELEÇÃO PÓS-GRADUAÇÃO - 2020/1		
Programa	PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA		
Curso	7049 - FÍSICA	**Nível**	MESTRADO ACADÊMICO
Modalidade	PRESENCIAL		

Dados pessoais dos membros da banca examinadora

Nome completo	Documento	Nasc	Titulação		
			Área	Local	Ano
JOSÉ RAFAEL BORDIN	009.555.210-33	1985	CIÊNCIAS	UFRGS	2013
MARCIA CRISTINA BERNARDES BARBOSA	366.388.030-34	1960	CIÊNCIAS	UFRGS	1988
ALEXANDRE DIEHL	447.261.120-15	1964	CIÊNCIAS	UFRGS	1997
VALDEMAR DAS NEVES VIEIRA	648.795.930-87	1972	CIÊNCIAS	UFRGS	2004

Membros da banca examinadora	Titulo	Assinatura
009.555.210-33 - JOSÉ RAFAEL BORDIN	DOUTORADO	
366.388.030-34 - MARCIA CRISTINA BERNARDES BARBOSA	DOUTORADO	
447.261.120-15 - ALEXANDRE DIEHL	DOUTORADO	
648.795.930-87 - VALDEMAR DAS NEVES VIEIRA	DOUTORADO	

Ao(s) 17 dia(s) do mês de Agosto de 2021 os membros acima nomeados para a defesa da DISSERTAÇÃO do estudante VINICIUS FONSECA HERNANDES matriculado no PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA, consideram APROVADO, estabelecendo o título definitivo da DISSERTAÇÃO como sendo "**Aprendizado de máquina aplicado para classificação de fases em sistemas de matéria mole**", e estabelecendo um prazo máximo de 30 dia(s) para as correções e entrega da versão definitiva.

Eu, José Rafael Bordin, atesto que o(s) membro(s) da banca listado(s) acima sem assinatura participaram da sessão de forma remota.

Presidente da Banca

AGRADECIMENTOS

Agradeço ao meu orientador, José Rafael Bordin, por acreditar em uma relação de orientação saudável, estimulante e amigável, pelas discussões científicas e pelos conselhos de vida; ao meu coorientador, Mario Lucio Moreira, com o qual venho aprendendo, sobre ciência e sobre a vida, desde a graduação; e a todos os professores da minha jornada acadêmica.

Agradeço aos meus colegas de grupo, do Bordin Lab, por me proporcionar um ambiente (virtual) de aprendizado e colaboração, e pela amizade.

Agradeço à minha companheira, Paola, por sempre me apoiar e me ajudar, por não ter enjoado de mim durante o isolamento, e por todo o carinho e a amizade.

Agradeço minha família por toda a assitência, por apoiar minhas escolhas e por acreditar na importância da educação e da ciência.

Agradeço à Universidade Federal de Pelotas e ao Programa de Pós Graduação em Física pela formação de qualidade e gratuita.

Agradeço à CAPES, pela concessão de bolsa, e ao CNPq e à FAPERGS pelos fomentos para pesquisa.

RESUMO

FONSECA HERNANDES, Vinicius. **Aprendizado de máquina aplicado para classificação de fases em sistemas de matéria mole**. Orientador: José Rafael Bordin. 2021. 101 f. Dissertação (Mestrado em Física) – Instituto de Física e Matemática, Universidade Federal de Pelotas, Pelotas, 2021.

Caracterizar as diferentes fases em sistemas da Matéria Mole é um desafio encontrado em muitos problemas na interface entre Física e Química – e o desafio se torna ainda maior para fluidos polimórficos. Especificamente, fluidos que apresentam fases vítreas, como a água, podem ter, além de múltiplas fases sólidas, mais de uma fase líquida ou vítrea, e até mesmo apresentar um ponto crítico líquido-líquido. Assim, propomos neste trabalho um algoritmo, baseado em redes neurais, para analisar o comportamento das fases de um fluido de caroço amolecido que possui polimorfismo na fase líquida, ponto crítico líquido-líquido e fase amorfa, similar à água. Também aplicamos o algoritmo para analisar uma mistura de modelos de caroço amolecido de água e álcool. Para isso, combinamos e expandimos dois métodos baseados nos parâmetros de ordem orientacional para estudar misturas: o método de Boattini e coautores [*Molecular Physics 116, 3066-3075 (2018)*], proposto para sistemas binários de caroço duro, e o método proposto por Martelli e coautores [*The Journal of Chemical Physics 153, 104503 (2020)*], para estudar água na região superresfriada. Incluindo camadas de coordenação de longo alcance, para incluir os terceiros vizinhos, a rede neural treinada foi capaz de prever, com alto grau de precisão, as fases sólidas cristalinas, as fases fluidas e a fase amorfa para o fluido de caroço-amolecido puro e para as misturas água-álcool. Além disso, utilizando a informação sobre as populações de cada fase em um dado ponto do diagrama, pudemos analisar como a região amorfa metaestável se espalha ao longo do diagrama de fases, especificamente na região de líquido de alta densidade. Estes resultados complementam o observado previamente, aumentando o entendimento do comportamento de fluidos polimórficos superresfriados e amplia a compreensão sobre como solutos anfifílicos afetam o diagrama de fases.

Palavras-chave: Redes Neurais. Dinâmica Molecular. Soluções Aquosas.

ABSTRACT

FONSECA HERNANDES, Vinicius. **Applied machine learning to phase classification of soft matter systems.**. Advisor: José Rafael Bordin. 2021. 101 f. Dissertation (Masters in Physics) – Institute of Physics and Mathematics, Federal University of Pelotas, Pelotas, 2021.

Characterization of phases of soft matter systems is a challenge faced in many physicochemical problems. For polymorphic fluids it is an even greater challenge. Specifically, glass forming fluids, as water, can have, besides solid polymorphism, more than one liquid and glassy phases, and even a liquid-liquid critical point. In this sense, we apply a neural network algorithm to analyze the phase behavior of a core-softened mixture of core-softened CSW fluids that have liquid polymorphism and liquid-liquid critical points, similar to water. We also apply the network on mixtures of CSW fluids and core-softened alcohols models. We combine and expand two methods based on bond-orientational order parameters to study mixtures, applied to mixtures of hardcore fluids by Boattini and co-authors [*Molecular Physics 116, 3066-3075 (2018)*] and to supercooled water by Martelli and co-authors [*The Journal of Chemical Physics 153, 104503 (2020)*], to include longer range coordination shells. With this, the trained neural network was able to properly predict the crystalline solid phases, the fluid phases and the amorphous phase for the pure CSW and CSW-alcohols mixtures with high efficiency. More than this, information about the phase populations, obtained from the network approach, can help verify if the phase transition is continuous or discontinuous, and also to interpret how the metastable amorphous region spreads along the stable high density fluid phase. These findings help to understand the behavior of supercooled polymorphic fluids and extend the comprehension of how amphiphilic solutes affect the phases behavior.

Keywords: Neural Networks. Molecular Dynamics. Aqueous Solutions.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

A tarefa de obter um diagrama de fases (DF) de um sistema físico, onde cada fase que o sistema pode assumir é bem delimitada e depende de propriedades como volume, temperatura e pressão, geralmente é uma tarefa complexa, maçante e custosa. Utilizando simulações computacionais [1] podemos facilitar e agilizar esse trabalho, já que os parâmetros do sistema são facilmente controláveis e com precisão inigualável por um experimento. Além disso, em uma simulação, tem-se fácil acesso a todas as variáveis do sistema, como a posição individual de cada molécula.

Em especial, simulações de Dinâmica Molecular (MD, do inglês *Molecular Dynamics*) se mostram como uma alternativa eficiente para modelagem de sistemas em áreas como Física, Química e Biologia. Essa abordagem permite a análise de quantidades termodinâmicas, dinâmicas, estruturais e estatísticas, que podem facilmente ser obtidas a partir de propriedades calculadas na simulação, para melhor compreender como ocorre o processo de transição de fase [2]. Um exemplo clássico é o calor específico a volume constante que, analisado em função da temperatura, apresenta uma descontinuidade quando o sistema passa por uma transição de fase.

Por mais que o processo de caracterização de fases seja facilitado ao utilizar simulações computacionais, dependendo do sistema, a tarefa continua demandando uma quantidade de tempo significativa, já que diferentes parâmetros precisam ser calculados e analisados para definir somente um ponto de transição. Uma alternativa que vem sendo explorada há aproximadamente uma década é utilizar modelos de aprendizado de máquina (ML, do inglês *machine learning*) para realizar de forma autônoma a caracterização de fases, evitando a necessidade de cálculos e análise de um número extensivo de parâmetros, economizando tempo e poder computacional. Nesse sentido, aplicamos um modelo de ML para estudar sistemas formados por misturas binárias água-álcool. Nas próximas seções da Introdução, o sistema analisado é apresentado e, na sequência, será comentado sobre aprendizado de máquina e suas aplicações na Física.

1.1 Misturas água-álcool

A vida, como conhecemos, começou e evoluiu em soluções aquosas. Assim, não é um exagero afirmar que elucidar o comportamento de moléculas complexas em água, tanto a nível macro quanto microscópico, é um objetivo de importância para a ciência moderna [3, 4]. Embora o comportamento de moléculas biológicas complexas em água seja um problema muito complicado, com muitos sítios hidrofílicos e hidrofóbicos, podemos obter algumas informações de sistemas mais simples. Por exemplo, de misturas de água e álcoois de cadeia curta (ou seja, álcoois com uma cadeia carbônica curta, como metanol, etanol e 1-propanol), cuja maioria são miscíveis em água em qualquer concentração [5, 6] por diversas outras razões: (i) eles são de extrema importância em áreas da indústria como médica [7], alimentícia [8], transporte [9] e cuidados pessoais [10]; (ii) a estrutura molecular dos álcoois tem um radical orgânico, enquanto que a água possui um hidrogênio. Como consequência, álcoois não formam uma rede de ligações de hidrogênio completamente desenvolvida; (iii) por outro lado, a característica anfifílica, devido à presença de um grupo hidroxila e do radical orgânico, normalmente não-polar, permite a interação com um número muito grande de componentes orgânicos e não orgânicos, fazendo dos álcoois ótimos solventes, uma vez que a interação soluto-solvente é da mesma ordem de magnitude das interações solvente-solvente [11]. Além disso, (iv) a anfifilicidade da molécula faz de álcoois um ótimo modelo para investigar efeitos hidrofóbicos [12, 13].

Contudo, mesmo água pura constitui um sistema extremamente complexo, com mais de 70 anomalias conhecidas [14, 15]. Todavia, neste trabalho estamos interessados em outra característica da água: seu polimorfismo. A água possui, literalmente, dezenas de fases sólidas observadas experimentalmente e outras preditas computacionalmente [14]. Mas existe outra transição que está diretamente ligada com suas anomalias [16, 17]: a transição líquido-líquido. De fato, está se chegando ao consenso que água liquída é, na verdade, dois líquidos competindo para dominar o comportamento do sistema [18, 19]. Estas duas fases líquidas, uma de baixa densidade (LDL, do inglês *Low-Density Liquid*), onde as moléculas de água se organizam em uma estrutura tetraédrica, e uma de alta densidade (HDL, do inglês *High-Density Liquid*), caracterizada por uma estrutura tetraédrica mais distorcida, e com maior densidade local. Como consequência, são separadas por uma linha de coexistência que acaba em um ponto crítico líquido-líquido (LLCP, do inglês *Liquid-Liquid Critical Point*). A discussão sobre a existência ou não do LLCP vem sendo realizada desde a primeira observação teórica nos anos 90, e ganhou particular atenção na última década [20–31]. Por mais que provar experimentalmente a existência desse ponto crítico seja uma tarefa extremamente complicada, já que nessa região superresfriada metaestável, conhecida como "terra de ninguém", o sistema cristaliza rapidamente, estudos recentes

apontam para sua existência [32–34], especialmente pela existência de duas fases amorfas, uma de alta densidade e outra de baixa densidade. Esse impecilho experimental torna a análise computacional da água nessa região uma necessidade e uma série de trabalhos com esse intuito já foram realizados [18, 25, 31, 35].

Uma forma de reduzir a complexidade do sistema, mas ainda manter características como o polimorfismo e as anomalias, é utilizar uma abordagem baseada em fluidos de caroço atenuado (CS, do inglês *Core-Softened*). A competição entre duas conformações distintas, uma representada pelo caroço duro e outra, mais distante, por uma casca amolecida, fazem com que estes modelos reproduzam as anomalias e o polimorfismo que surge na água devido à competição entre dois líquidos [36–41]. Entretanto, é importante ressaltar que essa aproximação envolve o uso de potenciais isotrópicos sem direcionalidade e, consequentemente, não retratam realmente água [42].

Em um estudo recente, parte da Tese de Doutorado de Murilo Sodré Marques na UFRGS, foi utilizado o modelo do poço de caroço amolecido (CSW, do inglês *Core-Softened Well*), proposto por Franzese [43] para o estudo de misturas água-álcool. Apesar de matematicamente simples, o modelo possui anomalias tipo água e polimorfismo, conforme observado anteriormente por diversos autores [43–46], e corroborado por resultados recentes [47, 48]. Assim, mostra-se na figura 1 o DF do modelo CSW no regime super resfriado conforme obtido por Marques e coautores [48], acompanhado de gráficos da configuração das partículas para cada fase. Aqui, o diagrama é apresentado para pressão e temperatura em unidades reduzidas adimensionais, distinguiveis das variáveis reais por carregar um asterisco (*) sobescrito. Nessa região a água apresenta cinco fases distintas. Como já mencionado, duas delas são líquidas, uma de baixa densidade e uma de alta densidade, além de três fases sólidas. Uma fase sólida na qual as moléculas se organizam formando uma estrutura cúbica de corpo centrada (BCC, do inglês *Body-Centered Cubic*), para pressões mais baixas. Outra, na qual a elevação da pressão comprime os partículas, que se organizam então em uma estrutura hexagonal compacta (HCP, do inglês *Hexagonal Closed-Packing*). Por fim, aumentando ainda mais a pressão, perde-se qualquer organização, e obtém-se uma fase amorfa, sem estruturação aparente, no entanto, sólida por não apresentar difusão.

Embora tenham sido reproduzidos, em [48], os resultados obtidos para o modelo CSW puro, o foco foi estudar como a presença de um soluto anfifílico afetaria as anomalias e o polimorfismo. Assim, como soluto foram utilizados os modelos de álcool de cadeia curta propostos e estudados por Urbic *et al.* [45, 49–51]. Nesta aproximação a hidroxila é modelada como um sítio do tipo CSW, enquanto que a cadeia carbônica é composta por sítios representados por um potencial do tipo Lennard-Jones. Assim, o álcool é um polímero rígido e linear com 2 (metanol), 3 (etanol) ou 4 (1-propanol)

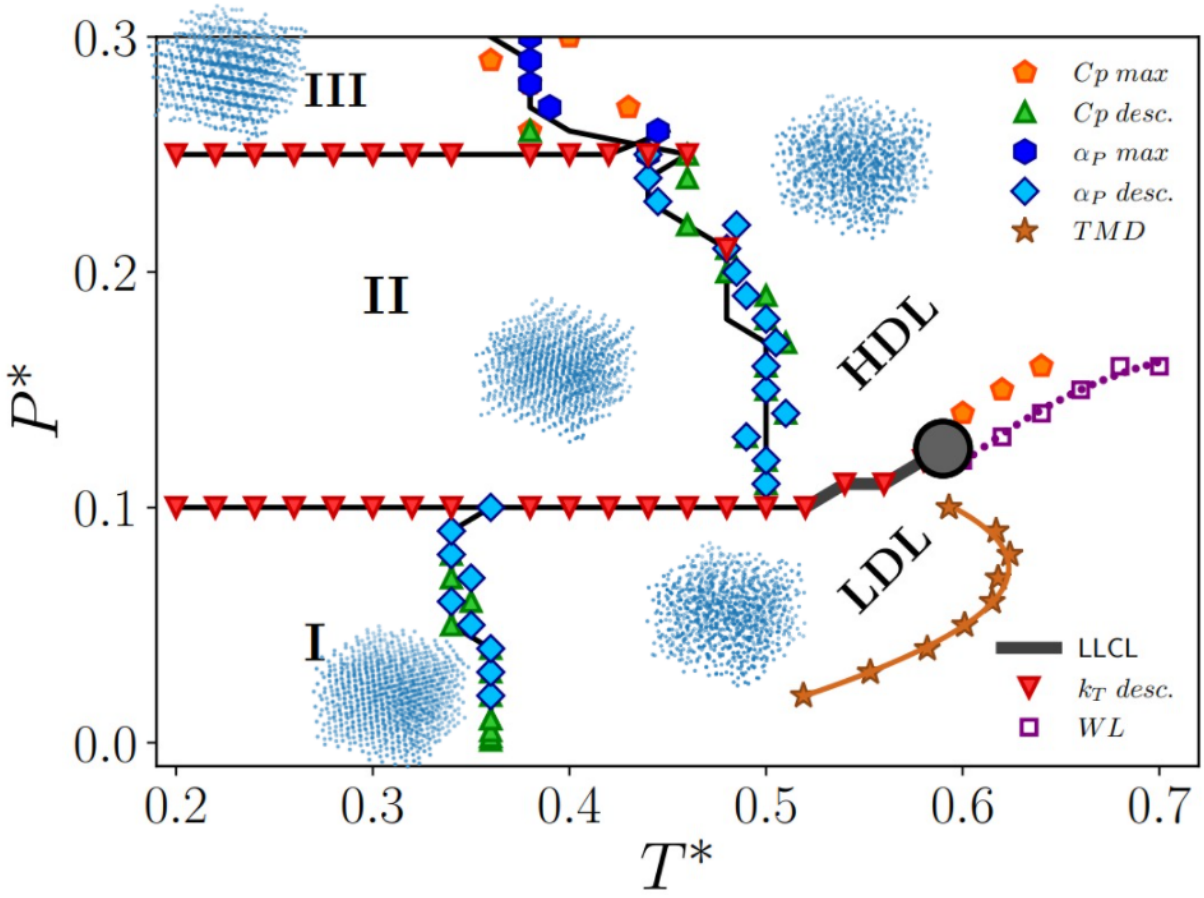


Figura 1 – Diagrama de Fases Pressão-Temperatura da água [48].

monômeros.

Ao introduzir álcool em água, notou-se que o diagrama de fase da figura 1 muda significativamente, dependendo de qual álcool está presente na mistura e em qual concentração. Na figura 2 exemplificamos isso apresentando nove diagramas de fases para misturas água-alcool, para o álcool podendo ser metanol, etanol ou propanol, e com concentrações 1%, 5% e 10%.

Assim como para água pura, além das fases HDL e LDL, o sistema apresenta três fases sólidas: BCC, HCP e amorfa (vítrea). Mesmo sendo metaestável, definimos no artigo original esta fase como sólida, pois não há difusão [48]. Ainda, para simplificar, a partir de agora essas fases sólidas serão denominadas fases I, II e III, respectivamente. Em especial, foi encontrado que a adição de álcool resulta na supressão da fase cristalina, favorecendo a fase III e mantendo a existência da transição líquido-líquido.

A análise termodinâmica realizada para encontrar os resultados obtidos por Marques *et al.* em [47, 48] demanda o cálculo extensivo de diferentes funções respostas, as quais precisam ser examinadas individualmente para encontrar a localização dos pontos de transição. Cada um dos símbolos coloridos (triângulos vermelhos e verdes, pentâgonos laranjas, etc.) nas figuras 1 e 2, correspondem a um desses pontos de

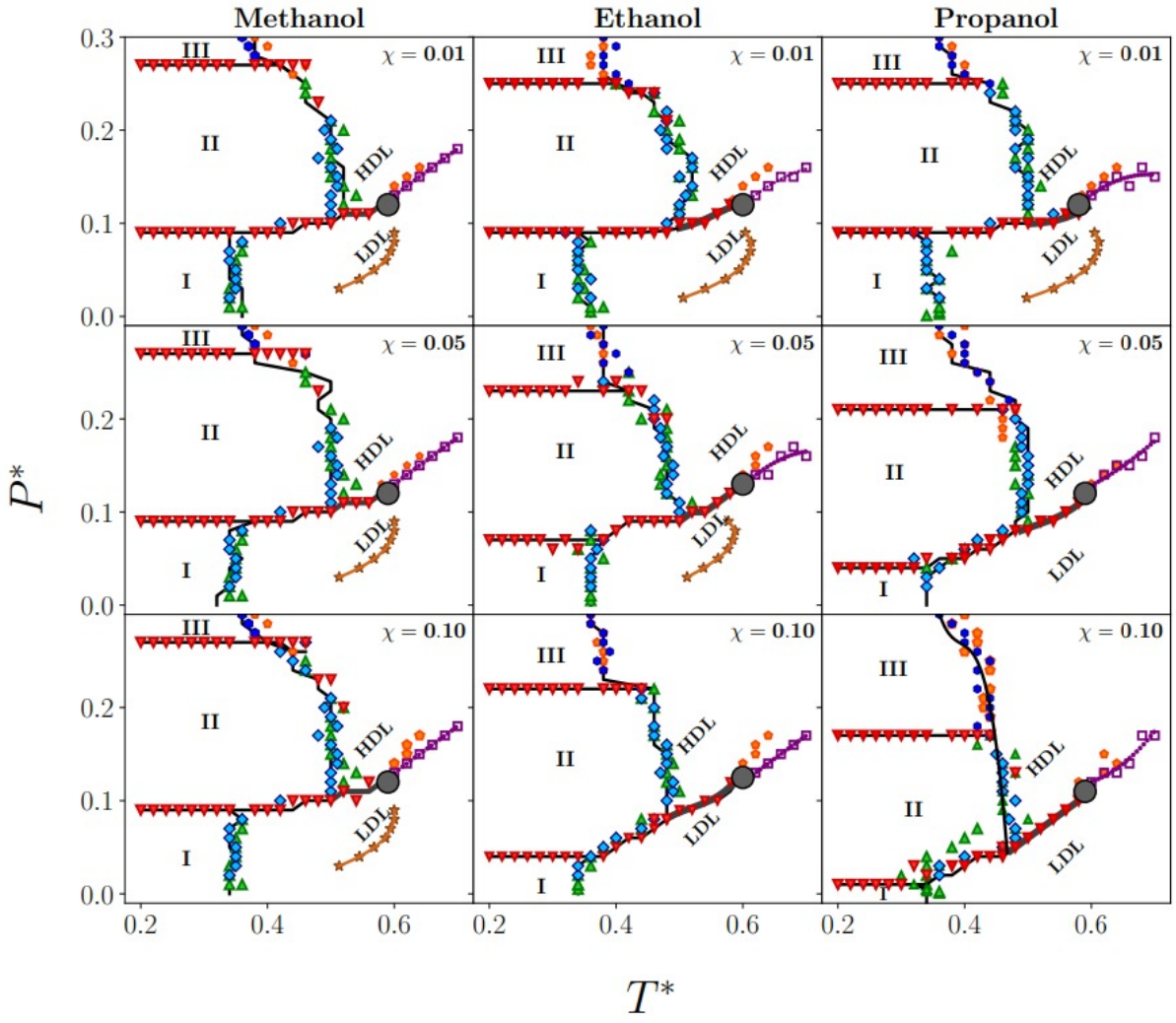


Figura 2 – Diagramas de fases para mistura água-álcool, para diferentes álcoois e concentrações [48].

transição de fase encontrados analisando parâmetros do sistema (expostos nas legendas das figuras) em função de uma certa propriedade. Além destes, parâmetros de ordem estrutural translacional e orientacional, além dos deslocamentos quadráticos médios e da difusão, foram calculados para definir as distintas fases. Para cada um dos diagramas, além da simulação, toda a análise termodinâmica-estatística para definir as fases precisa ser repetida, tornando facilmente o processo uma tarefa maçante – e este trabalho árduo e maçante foi exatamente a nossa inspiração. Desta forma, esta dissertação busca aplicar e expandir modelos de ML para classificar as fases em um sistema polimórfico.

1.2 Aprendizado de máquina

Aprendizado de máquina, na Ciência da Computação, é um tópico da área de inteligência artificial. De maneira geral, podemos classificar modelos de ML como algoritmos especializados no reconhecimento de padrões a partir de dados [52]. As primeiras execuções de ML ocorreram na metade do século XX, na forma de Redes Neurais (NN, do inglês *Neural Networks*), baseadas em neurônios e ligações entre esses, com uma clara inspiração no funcionamento do cérebro. O objetivo de modelos de aprendizado de máquina, como o nome sugere, é que o algoritmo consiga aprender certos padrões nos dados para que possa realizar sozinho alguma tarefa, como classificar uma imagem ou prever um valor de uma ação. Contudo, na sua concepção, as tarefas realizadas autonomamente pelas NNs ainda eram simples, como reconhecer padrões binários [53] ou geométricos, e classificação de primeiros vizinhos [54]; mesmo assim, essas implementações foram surpreendentes na época.

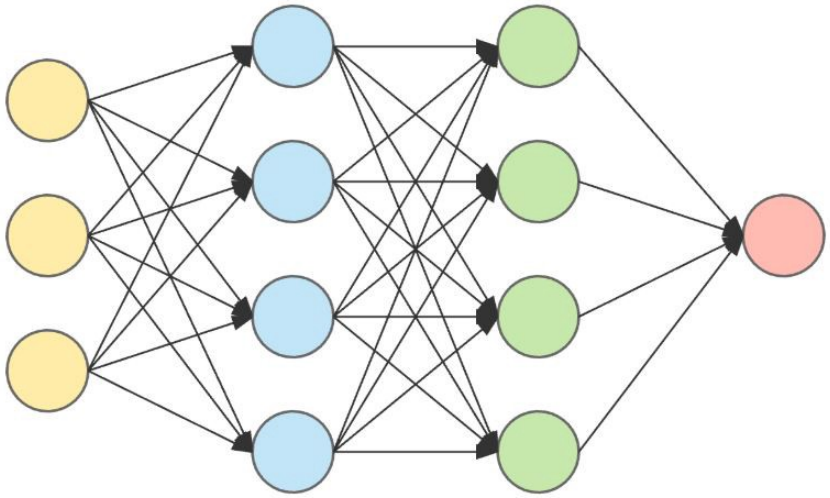
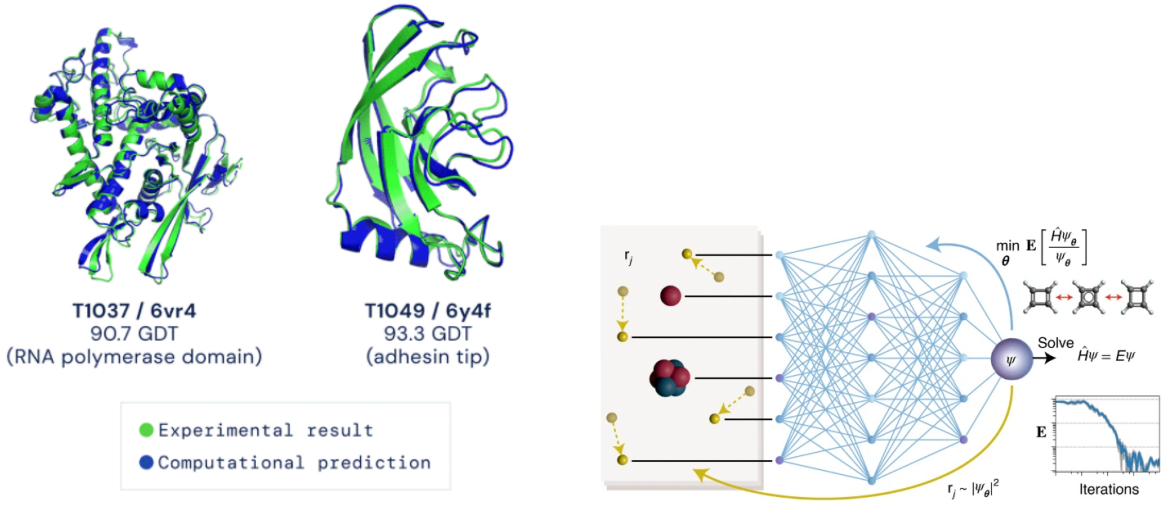


Figura 3 – Arquitetura de uma Rede Neural simples, com uma camada de entrada (amarela), camadas escondidas (azul e verde) e uma camada de saída (vermelho).

A área de ML só recebeu grande atenção a partir da década de 80, com a idealização de modelos mais complexos como o modelo de Hopfield [55, 56]. A partir desse momento, começaram a aumentar as semelhanças entre uma rede neural artificial e a rede neural biológica do cérebro, com diversos nós (ou neurões) alocados em camadas, e ligações entre eles, como esquematizado na figura 3. A evolução desses modelos acompanhou o aumento do poder computacional até o ponto no qual, hoje, conseguimos implementar redes com um grande número de camadas. Estes modelos ganharam a denominação de aprendizado de máquina profundo (do inglês *deep learning*) [57]. Desde a década de 90, com o crescimento exponencial da quantidade de dados disponíveis, a área de ML e, em especial, a área de aprendizado de máquina profundo, não para de crescer, com cada vez mais aplicações sendo encontradas. Nos últimos anos, a evolução da área de ML possibilitou avanços surpreendentes

em diversas áreas do conhecimento, como uma solução para o problema de *protein folding* [58], e os melhores resultados já vistos para solução da equação de Schrödinger [59], ilustrados nas figuras 4 (a) e (b), respectivamente.



(a) (b)

Figura 4 – (a) Comparação entre estrutura 3D de proteinas, preditas pelo modelo de ML, desenvolvido em [58], e os resultados experimentais. (b) Esquematização da metodologia aplicada em [59].

1.2.1 Processo de aprendizagem

Algoritmos de ML podem ser divididos em modelos de aprendizado supervisionado, nos quais o resultado esperado é conhecido previamente e o objetivo do modelo é relacionar os dados de entrada com os de saída, e modelos de aprendizado não supervisionado, onde não se conhece o resultado de antemão e espera-se que o modelo separe os dados de entrada em diferentes categorias de forma autônoma, e ainda modelos de aprendizado semi-supervisionado, que mesclam as duas técnicas. Outra classe de modelos é aquele de aprenziado por reforço, onde um agente aprende a realizar tarefas de forma a aumentar uma função recompensa [52]. De maneira geral, o objetivo de um algoritmo de aprendizado supervisionado é encontrar uma representação $f(\mathbf{x};\boldsymbol{\theta})$ que mapeia $f : \mathbf{x} \longrightarrow y$, onde $\mathbf{x}$ é um vetor que pertence à matriz $\mathbf{X}$ e y é um valor que pertence ao vetor $\mathbf{y}$, juntos constituindo o conjunto de dados $D(\mathbf{X}, \mathbf{y})$. Essa representação é construida de forma que os valores de $\boldsymbol{\theta}$ são aqueles que minimizam uma função erro (ou custo) $C(\mathbf{y}, f(\mathbf{X};\boldsymbol{\theta}))$. A matriz $\mathbf{X}$, denominada de matriz dos atributos, armazena os dados de entrada do modelo $(\mathbf{x})$, as variáveis independentes da equação f, enquanto o vetor $\mathbf{y}$, denominado vetor dos alvos ou dos rótulos, armazena os dados de saída (y), que são as variáveis dependentes de f [60].

Para alcançar esse objetivo, costuma-se dividir o conjunto de dados D em um conjunto de dados de treinamento D_{train} e um conjunto de dados de validação D_{val}.

Utilizando D_{train}, são buscados valores de $\hat{\boldsymbol{\theta}}$ que satisfaçam

$$\hat{\boldsymbol{\theta}} = \text{argmin}_{\boldsymbol{\theta}}\{C(\mathbf{y}_{train}, f(\mathbf{X}_{train}; \boldsymbol{\theta}))\}, \tag{1}$$

otimizando os valores de $\boldsymbol{\theta}$ em cada etapa do processo de treino, conhecendo previamente o resultado esperado y, e comparando o mesmo com o valor predito $\hat{y}$. Já na etapa de validação, utilizam-se os valores de $\hat{\boldsymbol{\theta}}$, sem atualizá-los mais, para calcular a função erro e assim verificar o quão eficiente é o modelo implementado. Cada modelo de ML apresenta diferentes características que são relacionadas à minimização da função erro. Um dos modelos mais conhecidos, e empregado neste trabalho, é a NN.

Mesmo que os primeiros modelos de NN tenham sido desenvolvidos ainda no século passado, devido ao avanço considerável do poder computacional a disposição, na última década a sua implementação cresceu consideravelmente, mostrando ótimos resultados nas mais diversas aplicações. Relacionando com a construção genérica de modelos de ML, a NN em si seria a função f a ser ajustada, minimizando $\hat{\boldsymbol{\theta}}$, de forma a encontrar a melhor relação para o conjunto de dados $D(\mathbf{X}, \mathbf{y})$. Especificamente, sua implementação consiste em armezanar em cada neurão (círculos coloridos da figura 3) o valor da soma dos valores dos neurões da camada anterior, multiplicados por um certo peso, somados por valores de *bias*. Ao pegar, por exemplo, o primeiro nó da segunda camada da figura 3, seu valor z^1 será dado por

$$z^1 = \sum_{i=1}^{3} w^i x^i + b^i, \tag{2}$$

com i indo de 1 a 3, já que a camada anterior é formada por 3 nós, cada um com valor x^i. w^i são os pesos, na figura 3 representados pelas ligações entre nós de diferentes camadas. Além disso, antes de ser passado para a próxima camada, o valor do nó é transformado por uma determinada função $\sigma_i(z^i)$, chamada de função de ativação, que é constantemente usada para adicionar não-linearidade ao modelo.

O mesmo processo é repetido para todos os neurões até chegar na camada de saída, que equivale ao vetor de alvos preditos $\hat{\mathbf{y}}$. A matriz de pesos W, formada pelos valores de pesos w^i somados aos valores de *bias* b^i, para cada camada j da rede, é equivalente aos parâmetros $\hat{\boldsymbol{\theta}}$, ou seja, armazena os valores a serem otimizados para minimizar a função custo. Em especial, os pesos são ajustados utilizando o processo de método do gradiente (em inglês *gradient descent*) e o conceito de *back-propagation*, onde tomando a direção inversa da direção da rede (da camada de saída para a de entrada) toma-se a derivada dos valores dos nós em relação aos pesos, fazendo com que todo o processo de aprendizado corresponda a uma trajetória para a função erro no espaço multi-dimensional dos pesos, que evolui sempre para a direção com maior variação, buscando o melhor mínimo local [61, 62]. Uma representação

gráfica de uma trajetória da função erro buscando um mínimo local, em função de dois parâmetros de uma rede neural, é exposto na figura 5, onde o gradiente de cores é utilizado para exemplificar o valor do gradiente em diferentes pontos da superfície [63]. Com a advento de NNs e modelos mais complexos, a aplicação de ML nas ciências vem crescendo constantemente e, em especial na Física, tem auxiliado em estudos em todos os campos.

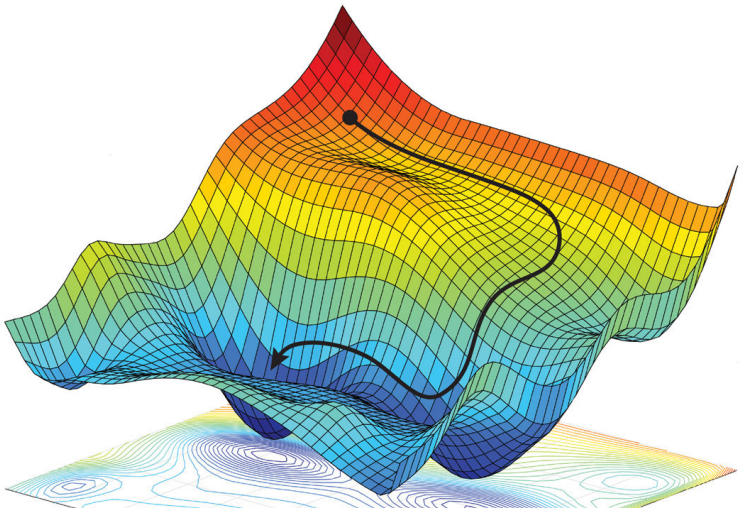


Figura 5 – Gráfico estilizado do método do gradiente [64].

1.2.2 Aplicações de aprendizado de máquina na Física

Assim como no resto da comunidade científica, a utilização de modelos de aprendizado de máquina em pesquisas na área de Física cresce ano a ano. Na figura 6 está apresentada a porcentagem de *pre-prints* publicados no repositório *arXiv*, na seção de Física, que contém o termo "*machine learning*" em seu resumo, de 2012 a 2020, em relação ao número total de artigos publicados na mesma seção. A porcentagem de 2020 corresponde a um total de 78857 trabalhos.

Nos *pre-prints* publicados no *arXiv* encontramos aplicações de ML em praticamente todas as áreas da Rísica, como em Rísica de Partículas [65], em simulações moleculares e atomísticas [66, 67], na parametrização de campos de força [68–70], na calibragem de computadores quânticos [71, 72], no estudo de *self-assembly* em moléculas [73–76], entre outros [77]. Uma dessas aplicações, útil principalmente na área de matéria condensada, diz respeito à classificação de fases da matéria. Modelos de aprendizado de máquina já vêm sendo utilizados para classificar fases dos mais variados tipos de sistemas, usando aprendizado não-supervisionado para encontrar transições de fase topológicas em modelos de Ising calibrados [78] e para classificar estruturas cristalinas com *autoencoders* [79], ou usando redes neurais convolucionais para detectar imagens de fases magnéticas para o modelo de Ising ferromagnético [80], e em matéria condensada mole, para sistemas coloidais em duas e três dimensões [81].

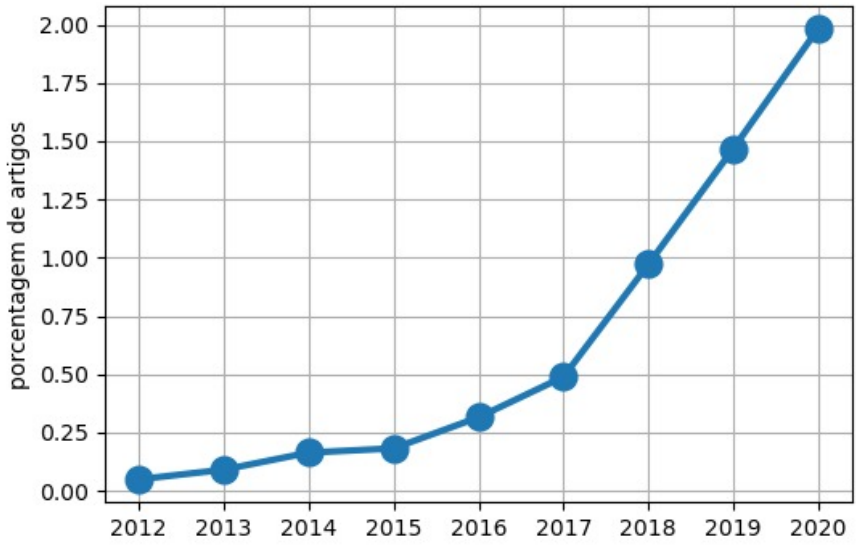


Figura 6 – Porcentagem de *preprints* publicados na seção de Física do repositório *arXiv* que contém o termo *machine learning* em seu resumo.

Para estudar água na região super resfriada, próxima ao segundo ponto crítico, diferentes abordagens com aprendizado supervisionado já foram empregadas, algumas delas utilizando modelos baseados em parâmetros de ordem [82], outras em funções de simetria [83], utilizando configurações obtidas por cálculos *ab-initio* [84, 85], ou ainda implementando redes com arquiteturas mais complexas, combinando diferentes métodos [86]. Outros trabalhos usaram aprendizado não supervisionado para estudar a relação entre estrutura e dinâmica para fluidos e vidros genéricos na mesma região [87, 88], que podem facilmente ser adaptados para água.

Contrastante com o caso da água pura, existe uma falta de trabalhos aplicando aprendizado de máquina para caracterizar as fases de soluções aquosas na região super resfriada, aplicação que pode auxiliar na compreensão desses sistemas ao se aproximar do LLCP. Logo, neste trabalho é implementada uma NN baseada em parâmetros de ordem orientacionais [89] e suas médias [90], como feito por Martelli *et al.* para água pura [82], e ainda usando as médias das médias, buscando adicionar ao modelo informação referente a estruturação da terceira camada de vizinhos. A rede se baseia também no método desenvolvido por Boattini e colaboradores [91] para distinguir diferentes tipos de moléculas da mistura binária. Espera-se que o modelo consiga classificar de forma precisa as fases presentes nas misturas estudadas em [48], conseguindo reproduzir os diagramas de fase expostos nas figuras 1 e 2, e ainda prover novas informações a respeito do polimorfismo presente na região superresfriada.

A estrutura da Dissertação é a que se segue. No Capítulo 2 são expostos os métodos utilizados para simular os sistemas considerados, e os detalhes da abordagem de aprendizado de máquina. No Capítulo 3 são apresentados os diagramas de fase encontrados pela abordagem de ML, comparando-os com os diagramas de fase obti-

dos em [48], além de dados que permitem explorar localmente as estruturas próximas ao ponto crítico líquido-líquido. No Capítulo 4 é apresentada uma discussão final, descrevendo como as principais descobertas dessa pesquisa impactam o estado da arte na área de aprendizado de máquina aplicado à matéria condensada mole. Nos apêndices são expostos os algoritmos utilizados para os cálculos de parâmetros de ordem e para a implementação da abordagem de ML, além do *pre-print* do artigo resultante desse trabalho, submetido para publicação em revista científica.

2 METODOLOGIA

O sistema água-álcool foi estudado utilizando Dinâmica Molecular no ensemble NPT, método para o qual os principais fundamentos estão expostos na primeira seção deste capítulo. Na seção seguinte são apresentados os detalhes da abordagem utilizada para a simulação do sistema água-álcool, na subseção 2.2.1, e para a análise do sistema usando aprendizado de máquina, na subseção 2.2.2.

2.1 Método de Dinâmica Molecular

De maneira geral, o método de MD consiste em resolver numericamente a segunda lei de Newton,

$$m_i \ddot{\mathbf{r}}_\mathbf{i} = \mathbf{f}_\mathbf{i}, \tag{3}$$

para um sistema de N partículas, onde m_i é a massa da i-ésima partícula,

$$\ddot{\mathbf{r}}_\mathbf{i} = \frac{d^2\mathbf{r}_\mathbf{i}}{dt^2} \tag{4}$$

é a aceleração da i-ésima partícula, resultante da força

$$\mathbf{f}_\mathbf{i} = -\nabla V, \tag{5}$$

dada pelo potencial de interação V. Para recuperar a velocidade e a posição da i-ésima partícula é necessário integrar a aceleração uma e duas vezes, respectivamente.

Comumente, para a evolução temporal do sistema, com incrementos infinitesimais δt ao tempo, utiliza-se o algorítmo *velocity*-Verlet [1], que é uma adaptação do algorítmo de Verlet [92]. Esse processo consiste em evoluir temporalmente o sistema em determinadas etapas. Como resultado, obtém-se a posição no tempo $t + \delta t$ dada por

$$\mathbf{r}_\mathbf{i}(t + \delta t) = \mathbf{r}_\mathbf{i}(t) + \mathbf{v}_\mathbf{i}(t)\delta t + \frac{1}{2}\mathbf{a}_\mathbf{i}(t)\delta t^2, \tag{6}$$

e a velocidade no tempo $t + \delta t$ dada por

$$\mathbf{v_i}(t + \delta t) = \mathbf{v_i}(t) + \frac{1}{2}[\mathbf{a_i}(t) + \mathbf{a_i}(t + \delta t)]\delta t. \tag{7}$$

Esse procedimento de evolução temporal é obtido desde um tempo inicial $t = 0$ até um tempo máximo t_{max}. Contudo, a etapa onde $t = 0$ não é a primeira etapa da simulação. Antes, são implementados os passos de inicialização do sistema e de termalização do sistema. Esses passos são necessários para garantir que as posições e as velocidades iniciais das partículas não sejam variáveis determinantes no cálculo de propriedades de interesse (como energia, temperatura cinética, etc.).

A inicialização do sistema consiste na alocação das N partículas em uma caixa, comumente com posições e velocidades aleatórias. Já a termalização é realizada com as integrações numéricas, utilizando *velocity*-Verlet, n vezes. Neste estágio, é comum monitorar grandezas como a energia e a temperatura do sistema, para garantir que o equilíbrio térmico seja atingido. Após, durante a evolução temporal na etapa de produção dos resultados, se calculam as propriedades de interesse do sistema, até que o tempo máximo seja atingido.

Um resumo do algorítmo típico utilizado em MD é apresentado na forma de fluxograma na figura 7.

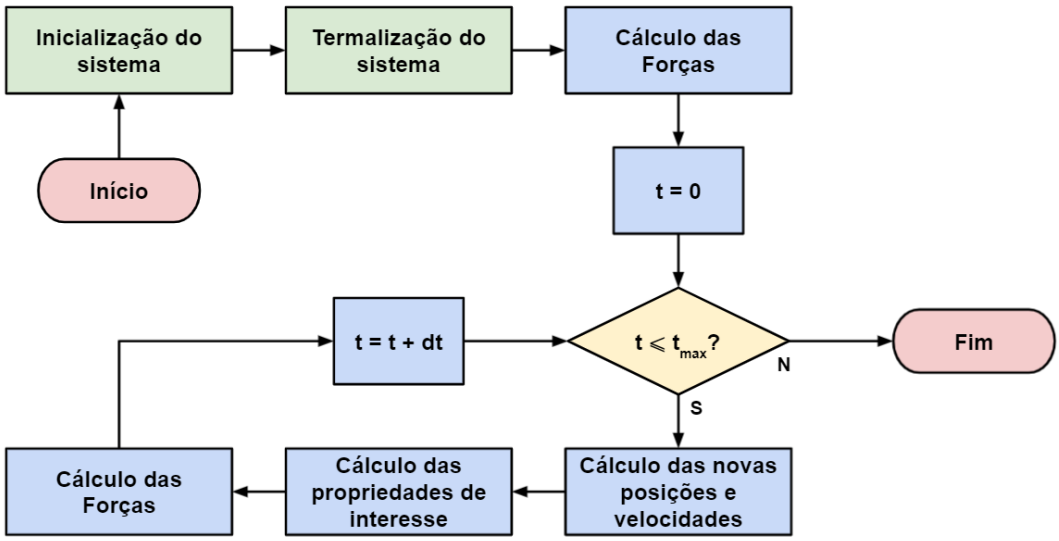


Figura 7 – Fluxograma do algorítmo típico de Dinâmica Molecular.

2.2 Modelo para a Mistura Binária Água-álcool

A modelagem do sistema água-álcool foi realizada utilizando o método de MD, exposto na seção anterior. A seguir são apresentados os detalhes da modelagem e, em seguida, a abordagem de aprendizado de máquina utilizada para implementar a

classificação autônoma de fases.

2.2.1 Modelagem do sistema

Moléculas de água foram tratadas como monômeros CSW na modelagem realizada, enquanto os difentes álcoois são tratados como cadeias poliméricas lineares. Metanol, $CH_3 - OH$, é simulado como um dímero (dois monômeros), etanol, $CH_3 - CH_2 - OH$, como um trímero (três monômeros) e 1-propanol, $CH_3 - CH_2 - CH_2 - OH$ como um tetrâmero (quatro monômeros). Especificamente, foi seguida a abordagem de Franzese *et al.* [43] para a água e aquela de Urbic e colaboradores [51], expandida por Marques e colaboradores [48], para os álcoois. A água e o grupo hidroxila (OH) são tratados como partículas de caroço amolecido, com um caroço duro de raio a e uma casca amolecida de raio $2a$. Essas partículas interagem pelo potencial

$$U^{CS}(r) = \frac{U_R}{1 + \exp[\Delta(r - R_R))]} - U_A \exp\left(-\frac{(r - R_A)^2}{2\delta_A^2}\right) + U_A \left(\frac{a}{r}\right)^{24} \tag{8}$$

conhecido como CSW, e os grupos carbônicos interagem entre sí, com o grupo hidroxila e com a água pelo potencial de Lennard Jones (LJ) adaptado

$$U^{LJ}(r) = \frac{4}{3} 2^{2/3} \epsilon \left[\left(\frac{\sigma}{r}\right)^{24} - \left(\frac{\sigma}{r}\right)^{6}\right], \tag{9}$$

com r sendo a distância entre duas partículas do sistema.

Na equação (8) tem-se $U_R = 2U_A$, $R_R = 1.6a$, $R_A = 2a$, $(\delta_A/a)^2 = 0.1$ e $\Delta = 15$, onde U_A é a profundidade do poço atrativo, U_R a altura do ombro repulsivo, R_A a posição do poço, R_R a posição do ombro, Δ é um parâmetro relacionado com a inclinação do potencial em R_R e δ_A é a variância da gaussiana centrada em R_A. Já na equação (9) tem-se $\sigma/a = 1$ e $\epsilon/U_A = 0.1$. Os valores específicos dos parâmetros foram escolhidos de modo a reproduzir o sistema de interesse, como feito por [51] e [43]. Para esses valores, o potencial CSW apresenta um poço atrativo na região próxima de $r = 2a$ e um ombro repulsivo próximo à $r = a$ (o que dá o nome para o potencial). Os potenciais em função da distância podem ser visualizados na figura 8(d). Nas figuras 8 (a), (b) e (c), estão esquematizadas as interações entre todas as partículas do sistema, para a mistura água-metanol, água-etanol e água-propanol, respectivamente.

Por mais que a equação (8) não represente realmente água, dada a falta de direcionalidade [42], a competição entre duas escalas, inerente do potencial, possibilita sua utilização para estudos de anomalias do tipo água, tanto em *bulk* quanto confinada, mostrando ótimos resultados [37, 38, 41, 93]. Uma das anomalias resultante do potencial é a existência do ponto crítico líquido-líquido [44], o que torna atrativa sua utilização para investigar o comportamento da água no regime super-resfriado. Na

equação (9) os diversos parâmetros dependem de qual álcool é considerado e ainda qual interação é considerada, gerando um grande número de combinações. Seus valores exatos, escolhidos para reproduzir da maneira mais precisa o comportamento dos diferentes alcóois, além de uma explicação mais detalhada da metodologia utilizada para a simulação podem ser encontrados em [48].

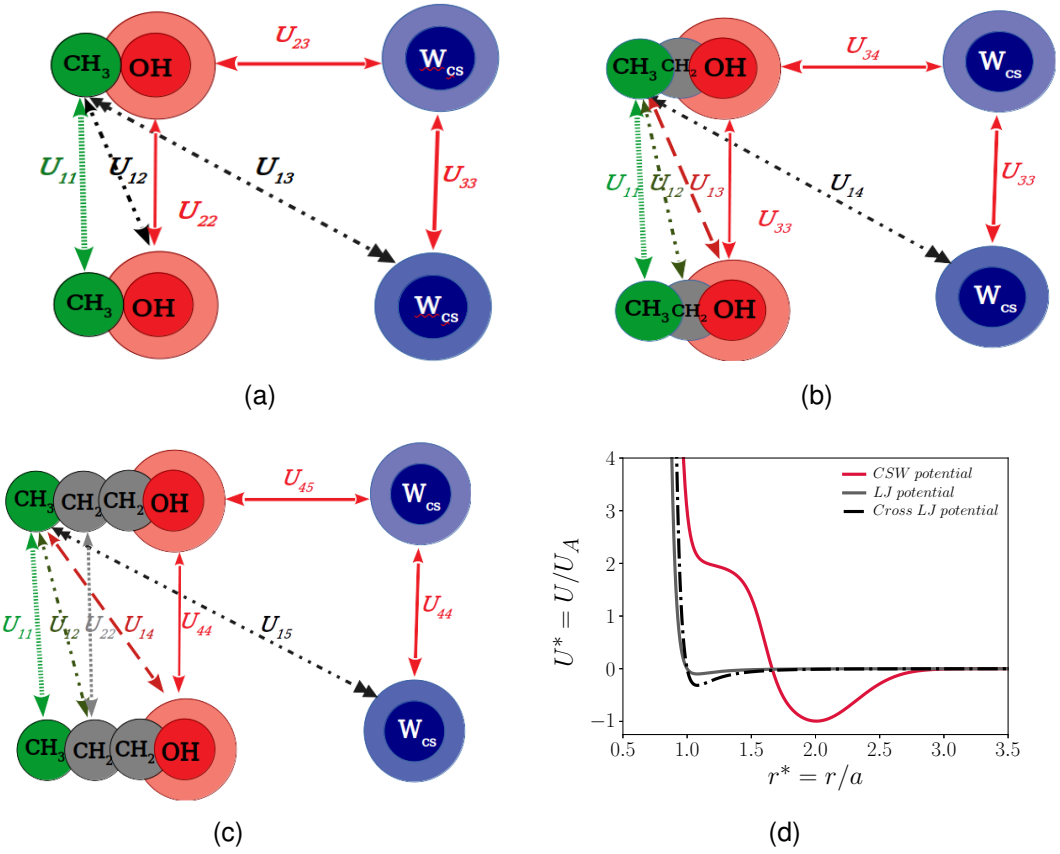


Figura 8 – Em (a), (b) e (c), o modelo de Franzese [43] e Urbic [45] para metanol, etanol e 1-propanol e água CSW com as diferentes combinações de interação e espécies. (d) Potencial de interação entre os diferentes sítios: o potencial CSW, o potencial de Lennard-Jones entre os diferentes sítios hidrofóbicos e com o solvente.

A modelagem dos sistemas, para as diferentes mistura água-álcool e para diferentes concentrações $\chi = 0.01, 0.05$ e 0.10 (ou seja para 1%, 5% e 10% de álcool), foi realizada utilizando o pacote ESPResSO [94,95]. A escolha por analisar somente o regime de baixa concentração é motivada pelo fato que esse é o regime onde características de interesse do sistema, como a presença de anomalias tipo água, são influenciadas significativamente por variações na concentração. As simulações foram realizadas no ensemble NPT (onde o número de partículas N, a pressão P e a temperatura T não variam) para $N = 1000$ moléculas. Especificamente, a pressão e a temperatura foram mantidas fixas utilizando dinâmica estocástica a pressão constante [96] e o termostato de Langevin, com parâmetros $\sigma_P = 0.0002$ e $m_P = 0.001$ para o barostato e $\gamma_0 = 1.0$ para o termostato. O número de moléculas de álcool depende da concentração de

forma que $N_{alc} = \chi N$. O número de moléculas de água é então $N_{ag} = N - N_{alc}$. O passo de tempo usado foi $\delta t^* = 0.01$. Para simular todos os sistemas que permitem reconstruir os diagramas de fase (figura 2) diferentes configurações (um sistema com determinados valores de pressão e temperatura) precisam ser simuladas. Especificamente, foram realizadas simulações para P^* variando de 0.01 a 0.30, com passo 0.01, e de 0.005 a 0.01, com passo 0.001, utilizando um processo de resfriamento, ou seja, diminuindo a temperatura T^* de 0.70 a 0.20, com passo de 0.02. Isso ocorre da seguinte maneira: para uma determinada pressão fixa, o sistema é inicializado na temperatura máxima e termalizado, em NVT (com um volume V fixo ao invés da pressão) rodando 5×10^6 passos de tempo. Então, 1×10^6 passos de tempo são rodados no NPT para equilibrar a pressão e a densidade, essa última dada por $\rho = N/\langle V_m \rangle$, onde V_m é o volume médio do sistema para P^* e T^* constantes. Após, mais 1×10^7 passos de MD são realizados, salvando grandezas para cálculo de resultados a cada 10^5 passos. Em seguida, o último estado do sistema é utilizado como estado de inicialização para simular a configuração à mesma pressão, para nova temperatura, e todo o processo é repetido, com a temperatura variando 0.02 entre cada conjunto de simulações, até atingir $T^* = 0.20$. Valores médios de quantidades como energia, densidade e temperatura instantânea foram salvos durante todo o decorrer da simulação para garantir que os sistemas simulados estão em equilíbrio. Todas as variáveis acompanhadas de asterisco estão em unidades reduzidas adimensionais relativas ao diâmetro do grupo hidroxila e a profundidade do poço atrativo: $T^* = k_B T/U_A$, $\rho^* = \rho a^3$ e $P^* = Pa^3/U_A$.

As grandezas obtidas na última etapa de simulação permitem estudar os sistemas, perceber o que há de diferente entre cada configuração e como determinadas quantidades termodinâmicas se comportam. No trabalho de Marques e colaboradores [48] uma série de grandezas foram calculadas para estudar a influência do tamanho de soluto nos sistemas, o comportamento da temperatura de máxima densidade e outras anomalias tipo água, aprofundando os estudos exclusivos para misturas água-metanol desenvolvidos em [47]. Nesses estudos mais genéricos e teóricos uma série de grandezas como entropia de excesso, parâmetro de ordem translacional, coeficiente de difusão, funções resposta, entalpia, etc., tiverem de ser analisados para compreender a fundo a física inerente das interações CSW e LJ. Neste presente trabalho, o interesse final é desenvolver um modelo de aprendizado de máquina capaz de classificar as diferentes fases dos sistemas, que portanto necessita de uma direta comparação com o DF. Dessa forma, como o objetivo desse trabalho consiste em desenvolver a abordagem e ML e comparar seus resultados (em especial, referentes aos DFs) com aqueles da abordagem termodinâmica, será dado enfoque na explicação de como esses diagramas foram obtidos, e somente os parâmetros necessários para encontrar as transições e construir o diagrama serão expostos a seguir.

Utilizando os valores de densidade e pressão para uma mesma temperatura, é

possível obter a compressibilidade isotérmica κ_T, dada por

$$\kappa_T = \frac{1}{\rho}\left(\frac{\partial \rho}{\partial P}\right)_T. \tag{10}$$

Já tendo os valores de densidade e temperatura para uma mesma pressão é possivel encontrar o coeficiente de expansão isobárica, dado por

$$\alpha_P = -\frac{1}{\rho}\left(\frac{\partial \rho}{\partial T}\right)_P, \tag{11}$$

e se a entalpia, dada por $H = U + PV$, também é conhecida, pode-se obter o calor específico a pressão constante da forma

$$C_P = \frac{1}{N}\left(\frac{\partial H}{\partial T}\right)_P. \tag{12}$$

Essas quantidades são chamadas de funções resposta, porque expõem a resposta do sistema ao mudar um de seus parâmetros. Por exemplo, ao plotarmos C_P em função de T, definimos pontos onde o sistema transiona de uma fase para outra quando C_P apresenta um máximo, para transições contínuas (segunda ordem), ou uma descontinuidade para transições descontínuas (primeira ordem). O mesmo ocorre para os outros parâmetros. Dessa forma, basta calcular as derivadas das equações (10), (11) e (12), utilizando diferenciação numérica, e traçar um gráfico em relação à pressão, para (10), e em relação à temperatura para (11) e (12), para encontrar os pontos de transição que permitem reconstruir os gráficos da figura 1 para água pura e da figura 2 para as diferentes misturas. Ao analisar a legenda dessas figuras, nota-se que os pontos coloridas que demarcam as regiões das cinco fases encontradas, são justamente máximos e descontinuidades das funções respostas expostas.

Outro dado obtido das simulações, útil para calcular diferentes parâmetros físicos e que será utilizado para montar o modelo de aprendizado de máquina, mas que não necessariamente é utilizado para construção do DF, é o estado das partículas (posições e velocidades) em diferentes instantes de tempo. Em especial, a posição de todas as partículas em um tempo específico (geralmente, o último passo de tempo da simulação, garantindo um sistema em equilíbrio) pode ser muito útil para encontrar simetrias e relações estruturais. Esses dados são salvos em um arquivo de trajetória, ou *snapshots*, no formato específico *xyz* que facilita sua manipulação e visualização das partículas, usando o software Ovito [97]. Nesse formato escreve-se na primeira linha o número N de partículas do sistema, na segunda uma *string* com informações do sistema, N linhas com quatro colunas, ou seja, uma linha por partícula com sua espécie e respectiva posição (x, y, z) em cada coluna. Um exemplo de *snapshot* para 3 partículas é exposto a seguir.

```
3
Comentario
X 1.543   2.628   0.432
X 3.703   4.686   8.348
X 5.138   9.413   6.160
```

Já na figura 9, obtida utilizando um arquivo de *snapshot* com o software Ovito, pode-se observar a posição de moléculas de água pura na fase I (cúbica de fase centrada). Esses dados de trajetória são a entrada para o nosso algoritmo de NN,

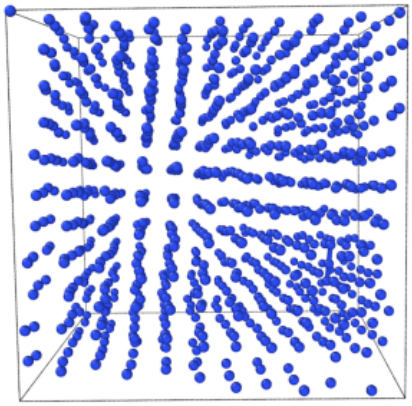


Figura 9 – Snapshot para água pura na fase I [48].

apresentado a seguir.

2.3 Abordagem de aprendizado de máquina

A obtenção de bons resultados ao utilizar um modelo de aprendizado de máquina depende intrinsicamente da utilização de bons dados [52]. Por isso, para construir um modelo capaz de classificar fases da matéria, os dados devem ser relacionados às fases. Neste trabalho segue-se a abordagem de utilizar parâmetros de ordem de ligação orientacional (BOOP, do inglês *Bond-Orientational Order Parameters*), a qual tem mostrado ótimos resultados para fases da água [82]. Os BOOPs são dados por

$$
\begin{aligned}
q_l(i) &= \sqrt{\frac{4\pi}{2l+1}\sum_{m=-l}^{l}|q_{lm}(i)|^2}, \\
\bar{q}_l(i) &= \sqrt{\frac{4\pi}{2l+1}\sum_{m=-l}^{l}|\bar{q}_{lm}(i)|^2}, \\
\bar{\bar{q}}_l(i) &= \sqrt{\frac{4\pi}{2l+1}\sum_{m=-l}^{l}|\bar{\bar{q}}_{lm}(i)|^2},
\end{aligned}
\tag{13}
$$

com

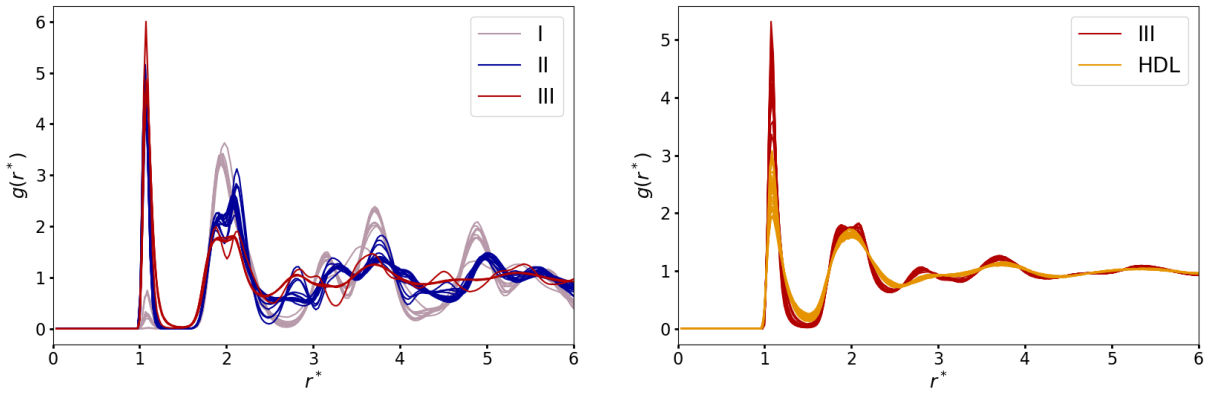
$$
\begin{aligned}
q_{lm}(i) &= \frac{1}{N_b(i)} \sum_{j=1}^{N_b(i)} Y_{lm}(\vec{r}_{ij}), \\
\bar{q}_{lm}(i) &= \frac{1}{N_b(i)+1} \sum_{k\in\{i,N_b(i)\}} q_{lm}(k), \\
\bar{\bar{q}}_{lm}(i) &= \frac{1}{N_b(i)+1} \sum_{k\in\{i,N_b(i)\}} \bar{q}_{lm}(k),
\end{aligned}
\tag{14}
$$

onde $N_b(i)$ é o número de primeiros vizinhos de uma partícula i, $Y_{lm}(\vec{r}_{ij})$ são os harmônicos esféricos para os números l e m e para duas partículas i e j separadas pela distância $\vec{r}_{ij}$.

Esses parâmetros são desenvolvidos para reproduzir relações de simetria de sistemas tridimensionais [89], tendo variações (diferentes valores de l) do mesmo parâmetro (q_l) relacionadas à diferentes simetrias. Por exemplo, um sistema com uma estrutura cúbica de face centrada terá um valor alto para q_6 e se a estrutura for hexagonal compacta, um valor alto para q_8. A segunda linha de (13) e (14) são os parâmetros médios, encontrados fazendo a média entre os parâmetros originais de uma determinada particula e seus primeiros vizinhos, que carregam informação sobre, aproximadamente, orientação até segundos vizinhos [90].

Neste trabalho são introduzidos os parâmetros médios-médios, expostos na terceira linha das equações (13) e (14). Esses parâmetros, de forma semelhante com os parâmetros médios, são encontrados tomando a média dos parâmetros médios para uma certa partícula e seus primeiros vizinhos, resultando em parâmetros que carregam informação sobre, aproximadamente, a orientação do sistema até terceiros vizinhos. Parâmetros médios-médios foram introduzidos para testar se a informação referente à terceiros vizinhos influencia a classificação das fases do sistema. A justificativa para sua utilização é encontrada ao analisar a função de distribuição radial, $g(r)$, de partículas CSW desses sistemas. Nota-se que seu comportamento varia na posição referente à terceira camada de vizinhos para diferentes configurações com a mesma fase. A função de distribuição radial é a probabilidade de encontrar uma partícula há uma distância r de uma partícula fixada. Como pode-se notar para o caso CSW puro, na figura 10(a) para diferentes pressões e temperatura fixa $T^* = 0.26$ - caso onde o sistema apresenta fases I, II e III, e na figura 10(b) para diferentes temperaturas e pressão fixa $P^* = 0.28$ - caso onde o sistema apresenta fases III e HDL.

Além dos BOOPs clássicos, costuma-se utilizar seus invariantes cúbicos w_l. Esses,



(a) (b)

Figura 10 – (a) Função de distribuição radial para água pura com $T^* = 0.26$. linhas cinzas representam isóbaras na fase I, azuis na fase II e vermelhas na fase III. (b) Função de distribuição radial para água pura com $P^* = 0.28$. Linhas vermelhas são isotermas na fase III e amarelas na fase HDL.

sua média e média das médias, são dados por

$$
\begin{aligned}
w_l(i) &= \frac{\sum_{m_1+m_2+m_3=0} \begin{pmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{pmatrix} q_{lm_1}(i) q_{lm_2}(i) q_{lm_3}(i)}{\left(\sum_{m=-l}^{l} |q_{lm}(i)|^2\right)^{3/2}}, \\
\bar{w}_l(i) &= \frac{\sum_{m_1+m_2+m_3=0} \begin{pmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{pmatrix} \bar{q}_{lm_1}(i) \bar{q}_{lm_2}(i) \bar{q}_{lm_3}(i)}{\left(\sum_{m=-l}^{l} |\bar{q}_{lm}(i)|^2\right)^{3/2}}, \\
\bar{\bar{w}}_l(i) &= \frac{\sum_{m_1+m_2+m_3=0} \begin{pmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{pmatrix} \bar{\bar{q}}_{lm_1}(i) \bar{\bar{q}}_{lm_2}(i) \bar{\bar{q}}_{lm_3}(i)}{\left(\sum_{m=-l}^{l} |\bar{\bar{q}}_{lm}(i)|^2\right)^{3/2}},
\end{aligned}
\tag{15}
$$

onde o termo entre parênteses nos denominadores é o símbolo $3j$ de Wigner.

Como sistemas reais (e até mesmo simulados) não são perfeitos, a utilização desses parâmetros pode não ser tão precisa, pois as distorções afetam seus valores, não conseguindo-se mais construir uma relação direta entre um parâmetro e a simetria correspondente. No entanto, utilizar extensivamente esses parâmetros (para vários valores de l) em um modelo de aprendizado de máquina tem mostrado ótimos resultados para distinção de fases, já que reunidos formam um conjunto de valores invariantes à rotação e a translação que pode caracterizar de forma unívoca a estrutura local de uma partícula do sistema. Nesse trabalho, seguindo a abordagem de Martelli *et al.* [82] utilizam-se valores de l variando de 3 a 12.

Ainda, como o sistema tratado é uma mistura binária, uma melhor caracterização

das estruturais locais pode ser obtida se forem utilizados BOOPs distintos para moléculas de álcool ligadas à outras moléculas de álcool e moléculas de álcool ligadas à moléculas de água, e o mesmo para o caso da água. Nesse sentido, adapta-se o modelo desenvolvido por Boattini e colaboradores [91], onde um modelo de aprendizado de máquina supervisionado é utilizado para classificar a estrutura local de um sistema binário, formado por partículas com diferentes raios. Para o caso das misturas água-álcool, a distinção não será entre partículas grandes e partículas pequenas como no trabalho de Boatinni *et al.*, mas sim entre moléculas de álcool e água. Dessa forma, para obter um sistema genérico para todas as misturas, considera-se somente o grupo comum entre os diferentes álcoois, o grupo hidroxila. Pegando como exemplo uma molécula i de soluto, ao invés de um único parâmetro $q_6(i)$, utiliza-se $q_6^{W-W}(i)$, $q_6^{A-A}(i)$ e $q_6^{W-A}(i)$, onde o primeiro termo é o parâmetro q_6 calculado considerando somente a interação com outras moléculas de solvente, o segundo termo será nulo pois é referente unicamente à moléculas de álcool, e o segundo termo é q_6 para a molécula de solvente, considerando somente moléculas de álcool como seus vizinhos. Para uma molécula de solvente, muda-se W por A nos parâmetros e os parâmetros referente somente à água ($W - W$) são nulos.

Utilizando os parâmetros expostos, e levando em consideração a abordagem para sistemas binários, obtém-se um vetor resultante $\mathbf{q}(i)$ para cada molécula i dado por

$$\begin{aligned}\mathbf{q}(i) = \Big(&\{q_l^{W-A}(i)\}, \{\bar{q}_l^{W-A}(i)\}, \{\bar{\bar{q}}_l^{W-A}(i)\}, \{q_l^{W-W}(i)\}, \{\bar{q}_l^{W-W}(i)\}, \{\bar{\bar{q}}_l^{W-W}(i)\},\\ &\{q_l^{A-A}(i)\}, \{\bar{q}_l^{A-A}(i)\}, \{\bar{\bar{q}}_l^{A-A}(i)\}, \{w_l^{W-A}(i)\}, \{\bar{w}_{l'}^{W-A}(i)\}, \{\bar{\bar{w}}_{l'}^{W-A}(i)\},\\ &\{w_{l'}^{W-W}(i)\}, \{\bar{w}_{l'}^{W-W}(i)\}, \{\bar{\bar{w}}_{l'}^{W-W}(i)\}, \{w_{l'}^{A-A}(i)\}, \{\bar{w}_{l'}^{A-A}(i)\}, \{\bar{\bar{w}}_{l'}^{A-A}(i)\}\Big),\end{aligned}$$

com l' assumindo somente os valores pares de l. Dessa forma, $l \in [3, 12]$ faz com que $\mathbf{q}(i)$ seja um vetor 135-dimensional que identifica unicamente cada partícula do sistema, conseguindo diferenciar entre moléculas de água e álcool, e que carrega informação sobre a estrutura local até terceiros vizinhos.

Computacionalmente, os parâmetros que compõem o vetor $\mathbf{q}(i)$ são obtidos utilizando o pacote em Python para pós-processamento de dados de simulações moleculares *freud* [98]. Primeiramente, é preciso importar esse pacote, juntamente com a biblioteca *numpy* [99], útil para processar vetores.

```
import numpy as np
import freud as fd
```

Então, para armazenar as posições de todas as partículas do último instante de tempo da simulação em vetores únicos para cada molécula, a partir de um certo arquivo $snap\_file$, utiliza-se o código a seguir.

```
1  N = int(np.loadtxt(file_snap, usecols = 0, max_rows = 1))
2
3  number_of_lines_in_snap = len(np.genfromtxt(file_snap, dtype = str, usecols
       = 0))
4  positions_all = np.loadtxt(file_snap, usecols = (1,2,3), skiprows=
       number_of_lines_in_snap - N, max_rows=N)
5  molecule = np.genfromtxt(file_snap, dtype=str, skip_header=
       number_of_lines_in_snap - N, max_rows=N, usecols=0)
6
7  positions = []
8  positions_N = []
9  positions_O = []
10
11 for i in range(len(positions_all)):
12     if(molecule[i] != 'F' and molecule[i] != 'S'):
13         positions.append(positions_all[i])
14         if(molecule[i] == 'N'):
15             positions_N.append(positions_all[i])
16         else:
17             positions_O.append(positions_all[i])
```

Na linha 1 é lida a primeira linha do arquivo, que corresponde ao número de partículas no sistema, e esse valor é armazenado na variável N. Em seguida, o número de linhas no arquivo é lido e armazenado na variável $number\_of\_lines\_in\_snap$, na linha 3. Na linha 4, utilizam-se as variáveis $number\_of\_lines\_in\_snap$ e N para garantir que somente a ultima configuração do *snapshot* é armazenada no vetor $positions\_all$, para as posições das partículas, e no vetor $molecule$, para o tipo de molécula. Em seguida, da linha 11 à 15, varre-se o vetor com as posições das partículas a fim de armazenar dados de posição de cada molécula em um vetor único. Especificamente, no vetor $positions$ são armezadas as posições de todas as moléculas do grupo hidroxila e água, no vetor $positions\_N$ somente as posições das moléculas de água e no vetor $positions\_O$ as posições das moléculas do grupo hidroxila. Em seguida, encontram-se os valores de comprimento em x, y e z que correspondem ao tamanho da caixa de simulação (L_x, L_y, L_z), a partir do vetor que armazena as posições de todas as partículas, fazendo

```
1  x,y,z = positions_all[:,0], positions_all[:,1], positions_all[:,2]
2  Lx, Ly, Lz = (x.max()-x.min()), (y.max()-y.min()), (z.max()-z.min())
```

com os quais é possível definir uma caixa que delimita o volume ocupado pelas partículas do sistema, que será um objeto da classe *Box()* dentro da biblioteca *freud*.

```
1  box = fd.box.Box(Lx, Ly, Lz)
```

Finalmente, os parâmetros de ordem podem ser calculados. Abaixo, mostra-se o procedimento para o cálculo de q_l para l arbitrátio, sem distinguir diferentes moléculas.

```
vor = fd.locality.Voronoi()

ngbs = vor.compute((box, positions)).nlist
q = fd.order.Steinhardt(l = l)
q_part = q.compute((box,positions), ngbs).particle_order
```

Na linha 1 acima é definido o objeto *vor* a partir da classe *Voronoi()*, que define que a tesselagem de Voronoi será utilizada para encontrar relações de vizinhança entre as partículas do sistema [100], armazenadas no vetor *ngbs*, na linha 3. Na linha 4 é criado o objeto *q* a partir da classe *Steinhardt(l = l)*, para o cálculo do parâmetro q_l com l específico, realizado na linha 5 para uma caixa *box* que contém moléculas de água e álcool (vetor *positions*) com a relação de vizinhança dada por *ngbs*. O valor do parâmetro q_l para cada partícula do sistema é então armazenado no vetor *q_part*. Para a obtenção dos parâmetros médios basta adicionar um argumento na classe *Steinhardt(l = l)*, substituindo a linha 4 e 5 acima por

```
qavg = fd.order.Steinhardt(l = l, avergare = True)
qavg_part = q.compute((box,positions), ngbs).particle_order
```

Já para encontrar os parâmetros médios-médios, em função dos parâmetros médios, pode-se fazer

```
ngbs_indices_list = [[i] for i in range(len(positions))]
for i, j in ngbs[:]:
    if(i == ngbs_indices_list[i][0]):
        ngbs_indices_list[i].append(j)

avg_aux = [[] for i in range(len(positions))]
qavg_avg_part = [0 for i in range(len(positions))]

for i in range(len(ngbs_indices_list)):
    for j in range(len(ngbs_indices_list[i])):
        avg_aux[i].append(qavg_part[ngbs_indices_list[i][j]])
    qavg_avg_part[i] = mean(avg_aux[i])
```

Na linha 1 armazenam-se o índice i de cada partícula do sistema no vetor *ngbs_indices_list*. Da linha 2 a 4 expande-se o vetor *ngbs_indices_list* para que os índices j dos vizinhos de uma dada partícula i, e o índice i sejam armazenados em um mesmo conjunto. Então, da linha 9 a 12, calcula-se a média do parâmetro médio para cada j-ésimo vizinho de i.

Então, para obter os parâmetros invariantes cúbicos, faz-se

```
if(int(l) % 2 == 0):
    w = fd.order.Steinhardt(l = l, wl = True)
    w_part = w.compute((box,positions), ngbs).particle_order
```

simplementes adicionando um argumento na classe *Steinhardt()*, e restringindo o cálculo para os valores pares de l. O mesmo procedimento mostrado anteriormente para

o cálculo de $\bar{q}_l$ e $\bar{\bar{q}}_l$ vale para $\bar{w}_l$ e $\bar{\bar{w}}_l$.

De forma semelhante pode-se fazer

```
ngbs = vor.compute((box, positions_N)).nlist
q = fd.order.Steinhardt(l = l)
q_part = q.compute((box,positions_N), ngbs).particle_order
```

para o cálculo dos parâmetros considerando somente ligações entre moléculas de água (vetor *positions_N*), ou

```
ngbs = vor.compute((box, positions_O)).nlist
q = fd.order.Steinhardt(l = l)
q_part = q.compute((box,positions_O), ngbs).particle_order
```

considerando somente ligações entre moléculas de álcool. A mesma correspondência faz-se presente para o cálculo dos parâmetros médios, médios-médios e invariantes cúbicos para as diferentes moléculas.

Com todos os parâmetros calculados, pode-se realizar a abordagem de aprendizado de máquina, que consiste então em ensinar uma rede neural a relacionar o vetor $\mathbf{q}(i)$ com uma determinada fase, ou seja, espera-se que, ao alimentar a rede, já treinada, com um vetor $\mathbf{q}(i)$, esta consiga prever em qual fase encontra-se a partícula i. Aplicando esse procedimento para todas as partículas de uma configuração e, em seguida, para todas as diferentes configurações, espera-se conseguir reproduzir todos os diagramas de fase expostos nas figuras 1 e 2.

A arquitetura da rede utilizada é dada por uma camada de entrada com 135 nós (já que como dado de entrada será utilizado um vetor 135-dimensional), três camadas escondidas com 180, 90 e 30 nós, respectivamente, e uma camada de saída com 5 nós, onde cada nó da camada de saída é relacionado a uma das 5 possíveis fases que os sistemas tratados podem assumir. Os nós são inicializados de forma uniforme utilizando o inicializador Glorot [101]. Como função de ativação para os nós da camada de entrada e das camadas escondidas, foi utilizada a função ReLu (*Rectifier Linear Unit*) e para a camada de saída a função *Softmax*, a qual confere uma relação probabilística para os nós da camada de saída, ou seja, o valor de um dos cinco nós da última camada será correspondente à probabilidade de uma partícula i com vetor $\mathbf{q}(i)$ associado estar naquela fase. A função erro escolhida foi *categorical cross-entropy* e *adam* [102] foi utilizado como otimizador.

A NN é implementada utilizando a API em Python *keras* [103] usando *TensorFlow* como *back-end* [104]. Esses são programas de alto nível que permitem construir, treinar e testar redes neurais, com diferentes arquiteturas e hiperparâmetros, de maneira fácil e prática. Em sua versão mais recente, *keras* está incluido na biblioteca *TensorFlow*, ou seja, para utilizar esses programas basta baixar a biblioteca e importá-la no algorítmo em Python com

```
import tensorflow as tf
```

O código que define a construção da NN é apresentado a seguir.

```
classifier = tf.keras.models.Sequential()
classifier.add(Dense(units = 180, use_bias=True, kernel_initializer='
    glorot_uniform', activation = 'relu', input_dim = 135))
classifier.add(Dense(units = 90, use_bias=True, kernel_initializer='
    glorot_uniform', activation = 'relu'))
classifier.add(Dense(units = 30, use_bias=True, kernel_initializer='
    glorot_uniform', activation = 'relu'))
classifier.add(Dense(units = 5, use_bias=True, kernel_initializer='
    glorot_uniform', activation = 'softmax'))
classifier.compile(optimizer = 'adam', loss = '
    sparse_categorical_crossentropy', metrics = ['accuracy'])
```

Nesse trecho o objeto *classifier* é construído a partir da classe *Sequential()*. Usando o objeto são adicionadas as camadas da NN e a rede é compilada. Após montar a rede, é preciso treina-lá, para que consiga aprender a relação entre o vetor $\mathbf{q}(i)$ e a fase da partícula i. Para compor o conjunto de dados para treinar a NN, 30 configurações foram escolhidas (2 configurações por fase das misturas de água-metanol, água-etanol e água-propanol, todas com concentração $\chi = 0.10$) e o vetor $\mathbf{q}(i)$ foi calculado para todas as $N = 1000$ partículas de cada uma das configurações. Além disso, como aplicamos aprendizado supervisionado, concomitantemente, constrói-se o vetor dos rótulos $\mathbf{y}$, formado pela fase de cada vetor $\mathbf{q}(i)$ correspondente. Como as configurações finais utilizadas para o cálculo dos parâmetros são de sistemas em equilíbrio, e ainda, como são escolhidas configurações afastadas de regiões de transições, infere-se que a configuração é uniforme, com as N partículas na mesma fase. Dessa forma, se for escolhida, por exemplo, a configuração com $P^* = 0.005$ e $T^* = 0.20$ para a mistura água-etanol, infere-se que 1000 partículas estão na fase I. Ainda, esses dados são divididos em um conjunto de treino e um de validação com proporção 90/10. Assim, o conjunto de dados de treino é formado por 27 mil pontos, com 135 *features* por ponto (todos os possíveis parâmetros que formam o vetor $\mathbf{q}(i)$ e um rótulo (a fase correspondente), enquanto o conjunto de validação é formado por 3 mil pontos.

Com o conjunto de treino pronto, a rede é treinada por 40 *epochs* (hiperparâmetro que definine o número de ciclos realizados pelo modelo durante o processo de aprendizado, ou ainda, quantas vezes o modelo utiliza todo conjunto de dados de treino para aprender) e usando um *batch* (hiperparâmetro que definine a quantidade de dados introduzida no modelo antes de atualizar os parâmetros internos) de tamanho igual a 32. No código isso é obtido fazendo

```
classifier.fit(X_train, y_train, batch_size = 32, epochs = 40)
```

onde X_{train} é o vetor de *features* do conjunto de treino, de tamanho (135x27000), e y_{train} é o vetor de rótulos do conjunto de treino, de tamanho (1x27000). Diferentes arquiteturas de NN, com diferentes hiperparâmetros, foram testadas, e a exposta acima

é aquela correspondente à que obteve a melhor acurácia. Nesse trecho do código, o objeto *classifier*, que antes estava com a estrutura da rede já compilado, ajusta os dados de entrada com os dados de saída, ou seja, encontra uma função que relaciona os dados de entrada com os dados de saída de forma que os parâmetros da função são aqueles que minimizam a diferença entre um valor predito para $y_{train}(i)$ e seu valor real.

Logo, o objeto *classifier* pode ser utilizado para classificar as fases de todas as partículas dos diferentes sistemas: todas as misturas com distintas concentrações, além de água CSW pura. Mais ainda, como o método utilizado permite a distinção da fase de cada partícula de uma configuração, ou seja retorna uma análise local, pode-se analisar a população de cada configuração. População é definida como o número de partículas em uma dada fase. Como exemplo, para uma configuração com todas as partículas na fase II, tem-se uma população (II) = 1000, já que $N = 1000$, e todas as outras populações são nulas. Essa análise permite verificar como as transições de fase ocorrem ao variar uma das quantidades termodinâmicas, possibilitanto um estudo focado na estruturação local do sistema.

3 RESULTADOS E DISCUSSÃO

3.1 Performance do Modelo de Aprendizado de Máquina

Diferentes parâmetros podem ser avaliados para analisar a viabilidade de um modelo de aprendizado de máquina. Algumas das métricas mais utilizadas incluem a acurácia, que pode ser pensada como a porcentagem de acertos do modelo, e o valor de erro (do inglês *loss*), que mensura a distância entre o valor predito pelo modelo e o valor real. De maneira geral, busca-se o valor de acurácia mais alto possível e o menor valor de erro associado possível. Contudo, deve-se tomar cuidado para evitar *overfitting*, quando o modelo não está realmente aprendendo uma relação entre os dados de entrada e saída, mas sim memorizando quais dados de entrada retornam um determinado valor de saída [52].

A NN construída neste trabalho tem como função de ativação para a camada de saída *softmax* que transforma um dado valor z_i de um conjunto $i \in K$, fazendo

$$\sigma(z_i) = \frac{\exp z_i}{\sum_{j=1}^{K} \exp z_j}. \tag{16}$$

Essa função faz com que os valores de saída retornem uma relação probabilística, ou seja, o valor de cada nó de saída é a probabilidade dos dados de entrada resultarem no valor associado aquele nó. Para o sistema tratado neste trabalho, cada nó carrega a probabilidade do vetor $\mathbf{q}(\mathbf{i})$ estar relacionado com a estrutura local de uma das cinco possíveis fases (I, II, III, LDL, HDL). A partir desses valores, considera-se que a fase predita pela NN é aquela associada ao nó da camada de saída com maior valor (maior probabilidade). Dessa forma, a acurácia é calculada como o número de fases corretamente classificados pela NN, dividida pela número total de pontos. Esse cálculo é realizado separadamente para o conjunto de treino e o conjunto de validação. A diferença entre eles é que, para o caso do treino, a rede utiliza o erro associado entre o valor predito e o valor real para aprender (processo de ajuste dos pesos), através do algoritmo de *backpropagation* otimizado *adam*. Já para o conjunto de validação, a rede retorna o resultado associado aos valores de entrada sem ajustar seus parâmetros internos. Essa diferença faz com que as métricas do conjunto de

validação sejam mais importantes para definidir a validade do modelo, já que para parte do processo de treino o algortimo conhece previamente o resultado esperado. Além disso, uma discrepância muito alta entre os valores de acurácia na etapa de treino e validação pode indicar *overfitting* do modelo.

A NN aplicada neste trabalho obteve acurácia de 99% na etapa de treino e de 99.3% na etapa de validação. O valor de erro foi aproximadamente 0.03 em ambas as etapas. Os valores de acurácia e erro em função das *epochs* são expostos na figura 11.

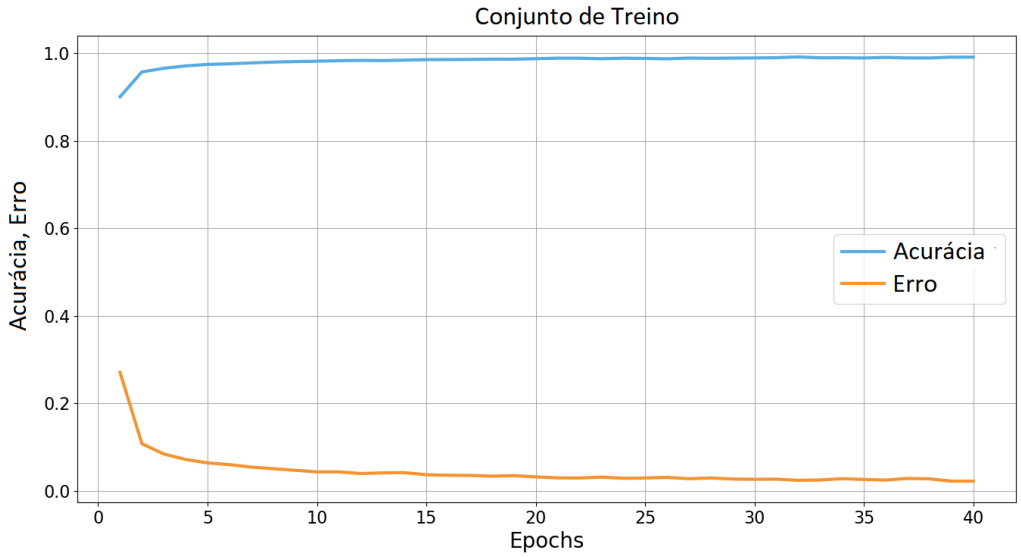


Figura 11 – Acurácia e erro na etapa de treino.

Nota-se que tanto a acurácia quanto o erro têm sua maior variação nas primeiras epochs e, no restante do processo de treino, a acurácia (o erro) aumenta (diminui) gradualmente. Isso indica que a rede aprende as relações entre os dados de entrada e saída já nos primeiros passos do treino e, a partir de então, passa a tentar memorizar os dados.

3.2 Classificação de Fases da Mistura Binária Água-álcool

Com a rede já treinada é feita a predição da fase de cada partícula de cada mistura com determinada concentração, além da água pura. Esses resultados são armezenados em um arquivo próprio para cada configuração (P, T), semelhante ao arquivo de *snapshots* adicionando a fase de cada partícula. Além disso, gera-se um arquivo para cada sistema, onde são especificados valores de P e T, a fase dominante, a população de cada fase, e o número total de partículas. Um exemplo das primeiras linhas do arquivo com essas informações, para a mistura água-etanol com concentração de etanol igual 10% é exposto a seguir.

```
# P      T     fase    I     II      III     LDL     HDL      N
0.20     0.6      4     0     0        0      10     990     1000
0.18     0.42     1     0     1000     0      0      0       1000
0.27     0.52     4     1     1        67     3      928     1000
0.16     0.52     1     0     1000     0      0      0       1000
0.10     0.3      1     0     1000     0      0      0       1000
```

A partir desses arquivo é possível reconstruir os diagramas de fase preditos pela NN, comparáveis com os encontrados usando a análise termodinâmica (figuras 1 e 2). Especificamente, na figura 12 é apresentado o DF da mistura água-etanol, com concentração de etanol $\chi = 0.10$.

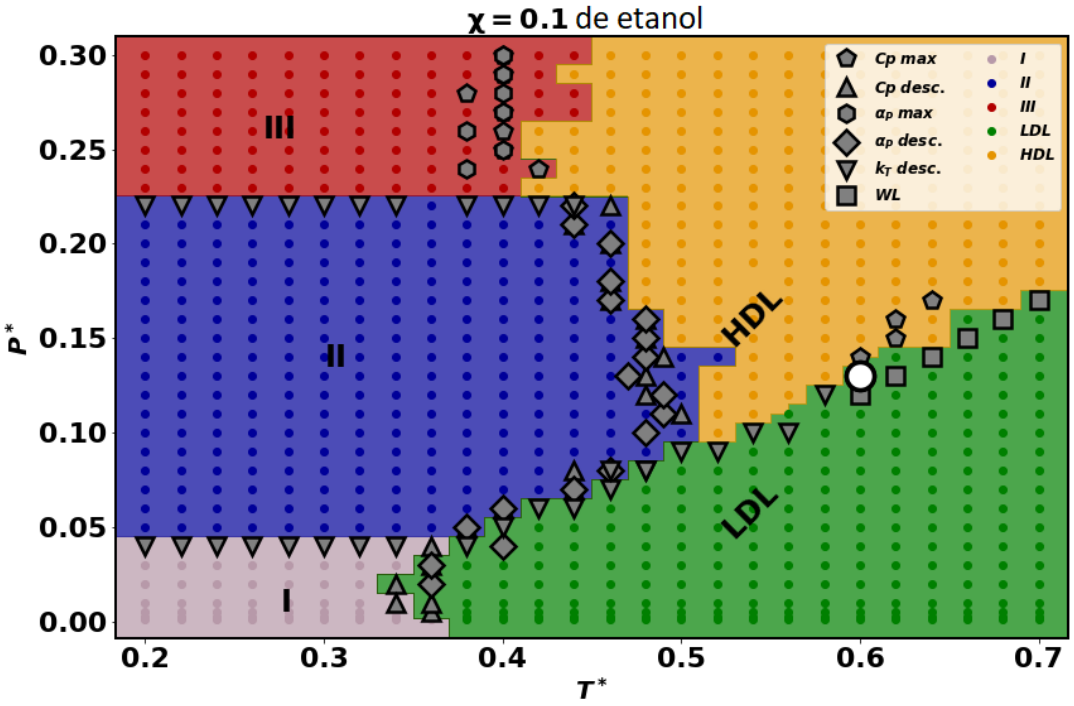


Figura 12 – Diagrama de fases obtido usando a abordagem de aprendizado de máquina, para a mistura água-etanol com concentração $\chi = 0.1$. A região com pontos cinzas corresponde a configurações classificadas como fase I, a região com pontos azuis como fase II, com pontos vermelhos como fase III, verde como fase LDL e amarelo como HDL. Os marcadores cinzas são os pontos de transição encontrados pela análise de funções resposta e o círculo branco é o ponto crítico líquido-líquido, conforme Marques e co-autores [48].

Cada ponto colorido da figura correspondente a uma configuração com determinado valor de pressão, temperatura e fase, onde a fase é encontrada considerando a fase dominante (com maior população), com cores diferentes para cada fase. Os marcadores cinzas (hexágonos, triângulos, etc.) são os pontos de transição encontrados analisando as funções resposta, especificamente a compressibilidade isotérmica κ_T, o coeficiente de expansão isobárica α_P e o calor específico a pressão constante C_P, e são os mesmos que se fazem presente nos diagramas das figuras 1 e 2. Como

esses pontos são encontrados pela análise usual que leva em consideração variáveis físicas do sistema, são considerados como referência para a NN, ou seja, o melhor resultados possível é aquele no qual as transições encontradas pela rede equivalem àquelas encontradas pela análise termodinâmica. Ainda, é mostrado o ponto crítico líquido-líquido, correspondente ao círculo branco na figura.

Nota-se pelo diagrama que as regiões das fases encontradas pela abordagem de aprendizado de máquina têm ótima correspondência com àquelas encontradas pela análise de funções resposta, com algumas divergências nas regiões de transição, particularmente na região de transição $III - HDL$. O mesmo comportamento é visto para as outras concentrações da mistura água-etanol, figuras 13 (a) e (b), água pura, figura 13(c), e para todas as concentrações das misturas água-metanol, figura 14 (a), (b) e (c), e água pronanol, figura 14 (d), (e) e (f).

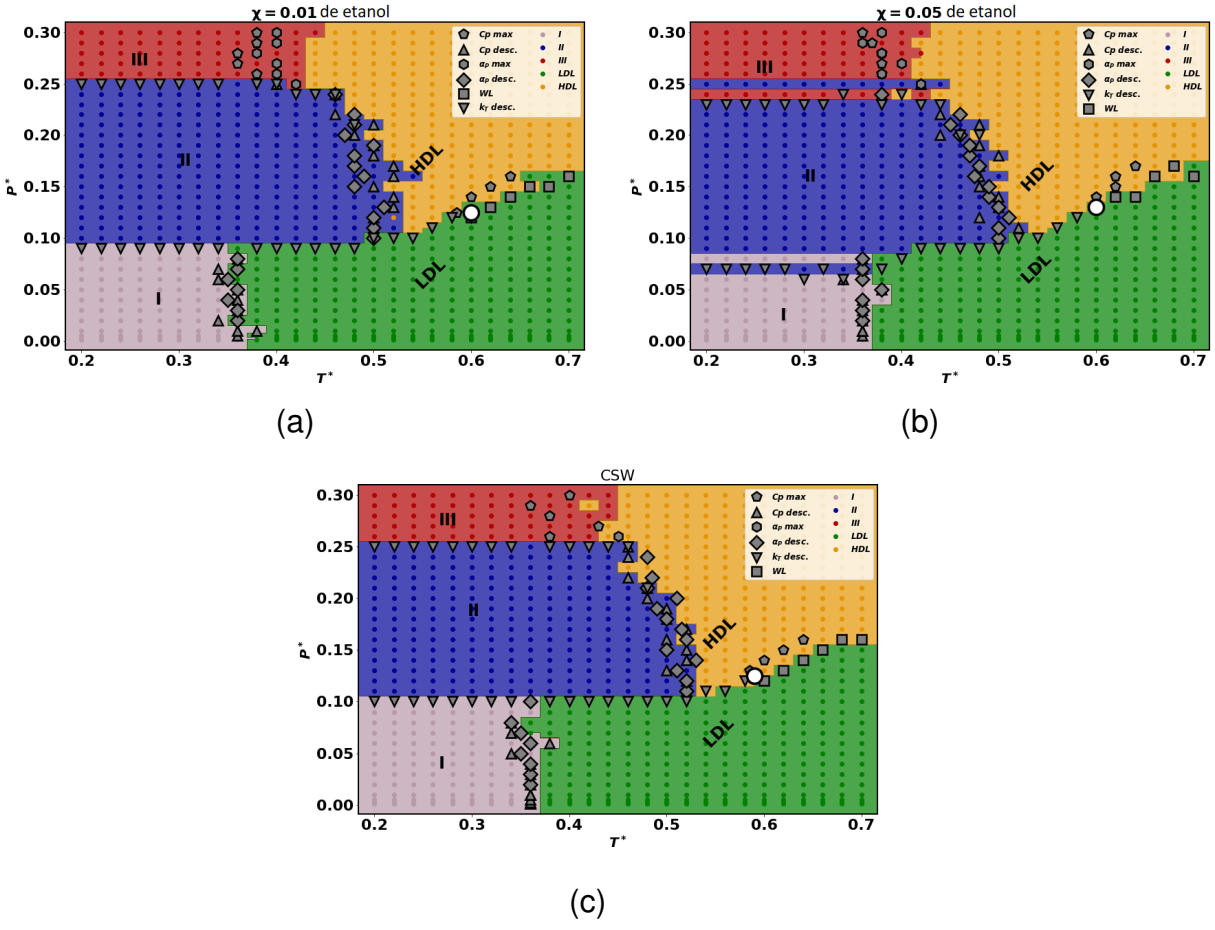


Figura 13 – Diagrama de fase, obtido usando a abordagem de aprendizado de máquina, para mistura água-etanol com concentração (a) $\chi = 0.01$ e (b) $\chi = 0.05$ e (c) para água pura. A região com pontos cinzas corresponde a configurações classificadas como fase I, a região com pontos azuis como fase II, com pontos vermelhos como fase III, verde como fase LDL e amarelo como HDL. Os marcadores cinzas são os pontos de transição encontrados pela análise de funções resposta e o círculo branco é o ponto crítico líquido-líquido [48].

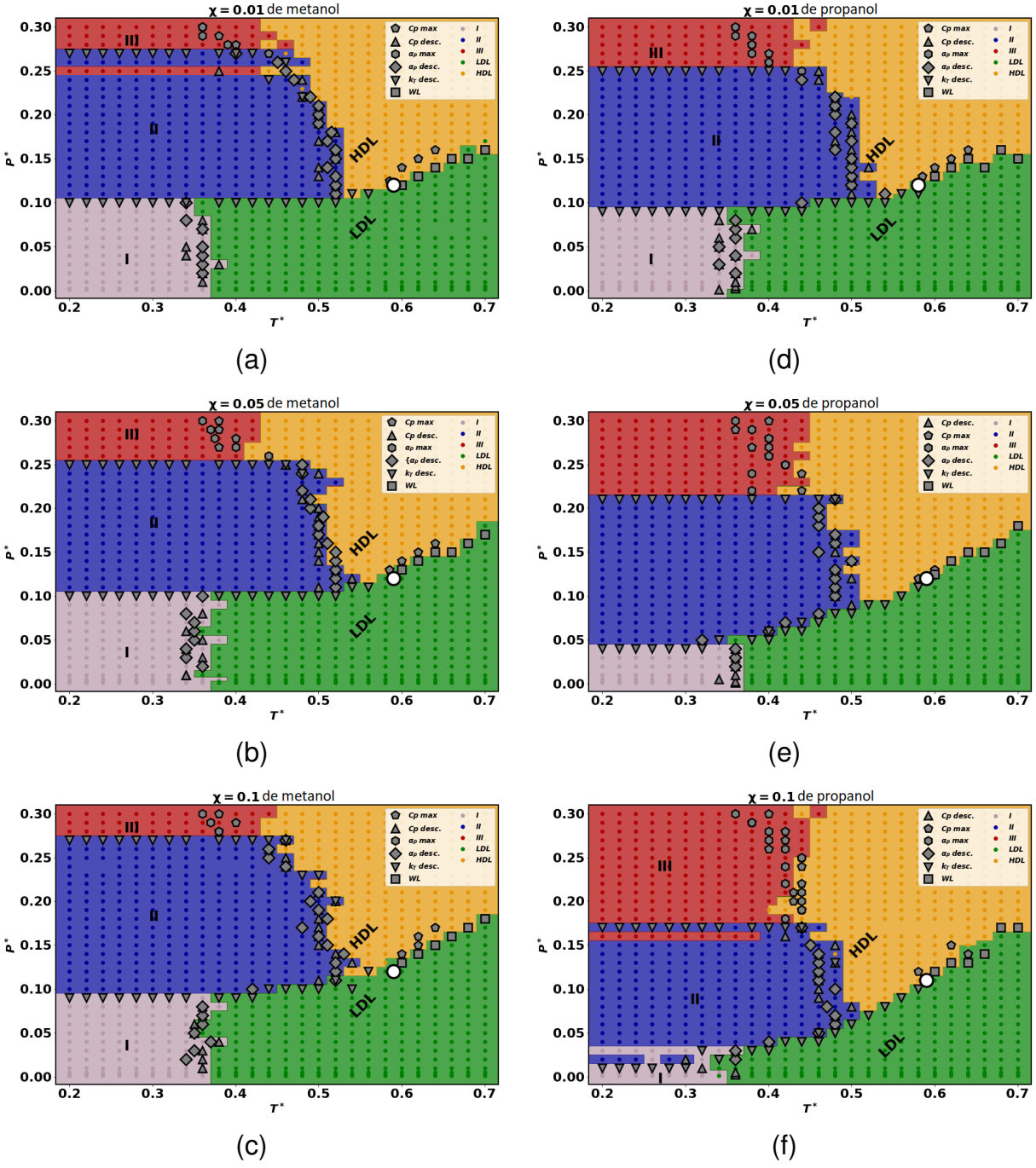


Figura 14 – Diagrama de fase, obtido usando a abordagem de aprendizado de máquina, para mistura água-metanol com concentração (a) $\chi = 0.01$, (b) $\chi = 0.05$ e (c) $\chi = 0.1$ e para mistura água-propanol com concentração (d) $\chi = 0.01$, (e) $\chi = 0.05$ e (f) $\chi = 0.1$. A região com pontos cinzas corresponde a configurações classificadas como fase I, a região com pontos azuis como fase II, com pontos vermelhos como fase III, verde como fase LDL e amarelo como HDL. Os marcadores cinzas são os pontos de transição encontrados pela análise de funções resposta e o círculo branco é o ponto crítico líquido-líquido [48].

Uma análise quantitativa da correspondência entre os dois métodos pode ser obtida através do cálculo da acurácia total de um sistema, definida como o número de

configurações com fase corretamente classificada pela NN, dividido pelo número total de configurações. Esses dados, para as diferentes misturas e água pura se encontram na Tabela 1.

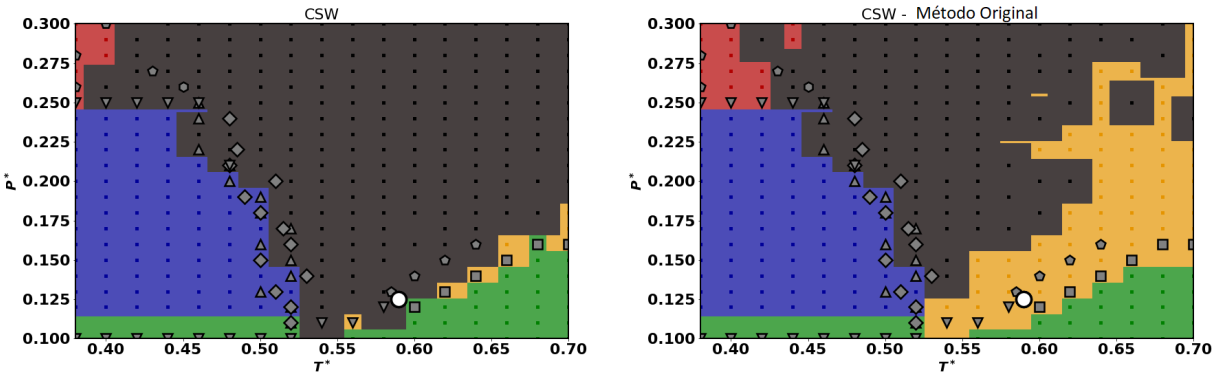
Tabela 1 – Acurácia total para água pura e todas as misturas.

	água pura	metanol			etanol			propanol		
χ	-	0.01	0.05	0.1	0.01	0.05	0.1	0.01	0.05	0.1
Acurácia	0.94	0.93	0.92	0.94	0.89	0.92	0.93	0.90	0.94	0.92

Mesmo com algumas isóbaras com fases classificadas de forma distinta, como nas figuras 13 (b), 14 (a) e (f), a acurácia total não mostra grande variação entre diferentes sistemas. Além disso, em todos os casos, a acurácia é consideravelmente menor do que aquelas encontradas nas etapas de treinamento e validação. Isso é atribuído aos pontos encontrados perto de transições de fase que foram classificadas de forma distinta. O conjunto de treinamento/validação é formado por configurações longe de transições para garantir uniformidade na fase das partículas individuais. Essa escolha faz com que a rede consiga aprender com êxito as características estruturais presentes no vetor $\mathbf{q}(i)$, mas, ao mesmo tempo, que tenha uma maior dificuldade em classificar fases nas regiões onde o vetor $\mathbf{q}(i)$ pode apresentar atributos compartilhados entre duas (ou mais) fases. Se configurações próximas a pontos de transição forem escolhidos para formar o conjunto de treinamento/validação, a acurácia total varia significamente e, de maneira geral, diminui. Essa constatação reforça a importância da escolha de bons dados para treinar modelos de aprendizado de máquina e, para o tipo de sistema analisado neste trabalho, isso significa utilizar configurações longe de pontos de transição. O valor mais baixo da acurácia também reforça que a NN aprende rapidamente as relações dos dados e, para passos seguintes da etapa de treino, tenta aumentar a acurácia ao memorizar dados (figura 11).

A divergência entre a fase amorfa e a HDL pode ser melhor explorada utilizando os dados de população. Pela abordagem termodinâmica, o ponto de transição dessa região pode variar se o sistema é simulado por um processo de aquecimento ou resfriamento, o que dificulta a obtenção de pontos de transições precisos. Também é importante frisar que a fase amorfa/vítrea é uma fase metaestável, aqui classificada como sólida por apresentar coeficiente de difusão próximo de zero, mas que ao mesmo tempo apresenta características muito semelhantes com a fase líquida de alta densidade, para o ordenamento de curto alcance [48]. Essas correspondências fazem com que para grandes regiões do diagrama de fase sejam encontradas configurações com fase metaestável, na qual a fase dominante foi classificada pela rede como fase HDL, mas o número de partículas classificadas como III não é nulo, e para a qual a rede não conseguiu distinguir de forma ideal a fase amorfa da fase líquida de alta densidade.

Para aprimorar essa distinção, levando em consideração a semelhança no ordenamento de curto alcance, neste trabalho expande-se a abordagem implementada por Martelli e colaboradores [82] para incluir parâmetros médios-médios, que carregam informação referente à aproximadamente terceiros vizinhos (última linha das eqs. (13) e (15)), ainda considerando diferentes parâmetros para diferentes moléculas, como proposto por Boattini *et al.* [91]. Na figura 15 é apresentado um enfoque do diagrama de fase, deixando explícita a fase metaestável, em preto, para o método utilizando paramêtros de ordem e suas médias (a), semelhante às abordagens desenvolvidas por [91] e [82], e incluindo parâmetros médios-médios (b).



(a) (b)

Figura 15 – Diagrama de fase na região $0.38 \leq \mathbf{T}^* \leq 0.70$ e $0.10 \leq \mathbf{P}^* \leq 0.30$ predito pela rede neural usando BOOPs e BOOPs médios (a) e incluindo parâmetros médios-médios (b) para água pura. A região com pontos azuis corresponde a configurações classificadas como fase II, a região com pontos vermelhos como fase III, com pontos verdes como fase LDL e amarelo como HDL. A região com pontos pretos é fase metaestável.

Comparando os dois métodos, nota-se que a inclusão dos parâmetros médios-médios diminui significativamente o tamanho da região com partículas na fase metaestável. Isso confirma que a informação referente à terceiros vizinhos é importante para a distinção entre a fase vítrea e a fase líquida de alta densidade. Também pode-se usar a mesma análise para estudar a influência da concentração de álcool na fase metaestável. Observa-se nas figuras 16, 17 e 18 que, para todas as misturas, a região delimitada pela fase metaestável diminui conforme aumenta a concentração de álcool, quando usado o método introduzido neste trabalho. O mesmo não ocorre usando somente BOOPs e BOOPs médios, caso para o qual não nota-se diferença significativa no tamanho da região metaestável. Esse resultado, além de deixar claro novamente uma melhor capacidade do modelo desenvolvido neste trabalho para diferenciação das fases vítreas e HDL, demonstra que baixas concentrações de álcool têm influência somente no ordenamento de longo alcance, enquanto concentrações mais elevadas influenciam predominantemente o ordenamento de curto alcance. A adição de

parâmetros médios-médios possibilita realizar essa análise, que concorda com [48], mostrando-se como uma alternativa mais genérica do que os métodos previamente desenvolvidos.

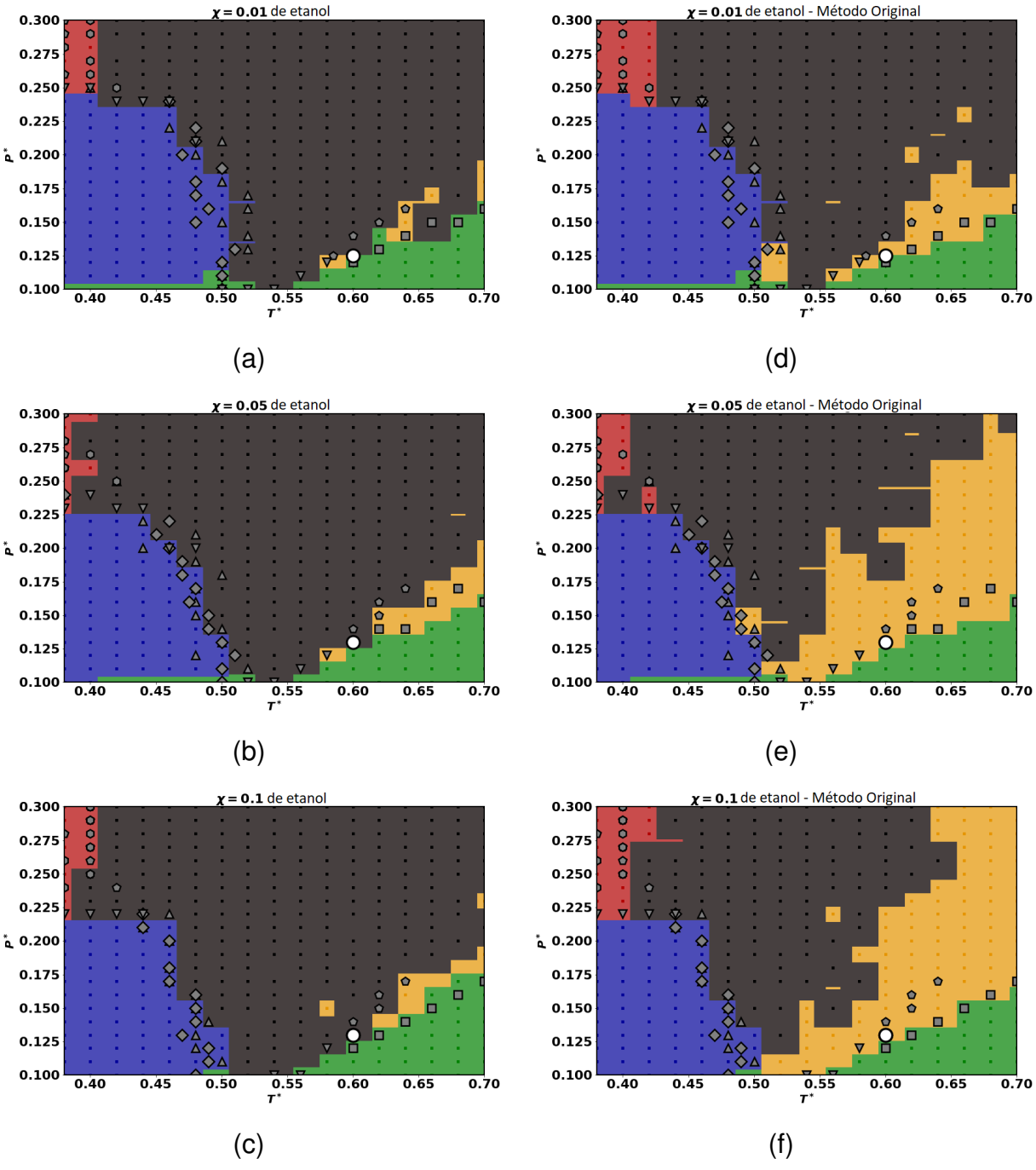


Figura 16 – Diagrama de fase na região $0.38 \leq \mathbf{T}^* \leq 0.70$ e $0.10 \leq \mathbf{P}^* \leq 0.30$, predito pela rede neural usando BOOPs e BOOPs médios para a mistura água-etanol com concentração (a) $\chi = 0.01$, (b) $\chi = 0.05$ e (c) $\chi = 0.1$ e incluindo parâmetros médios-médios para mesma mistura e concentrações (d) $\chi = 0.01$, (e) $\chi = 0.05$ e (f) $\chi = 0.1$. A região com pontos azuis corresponde a configurações classificadas como fase II, a região com pontos vermelhos como fase III, com pontos verdes como fase LDL e amarelo como HDL. A região com pontos pretos é fase metaestável.

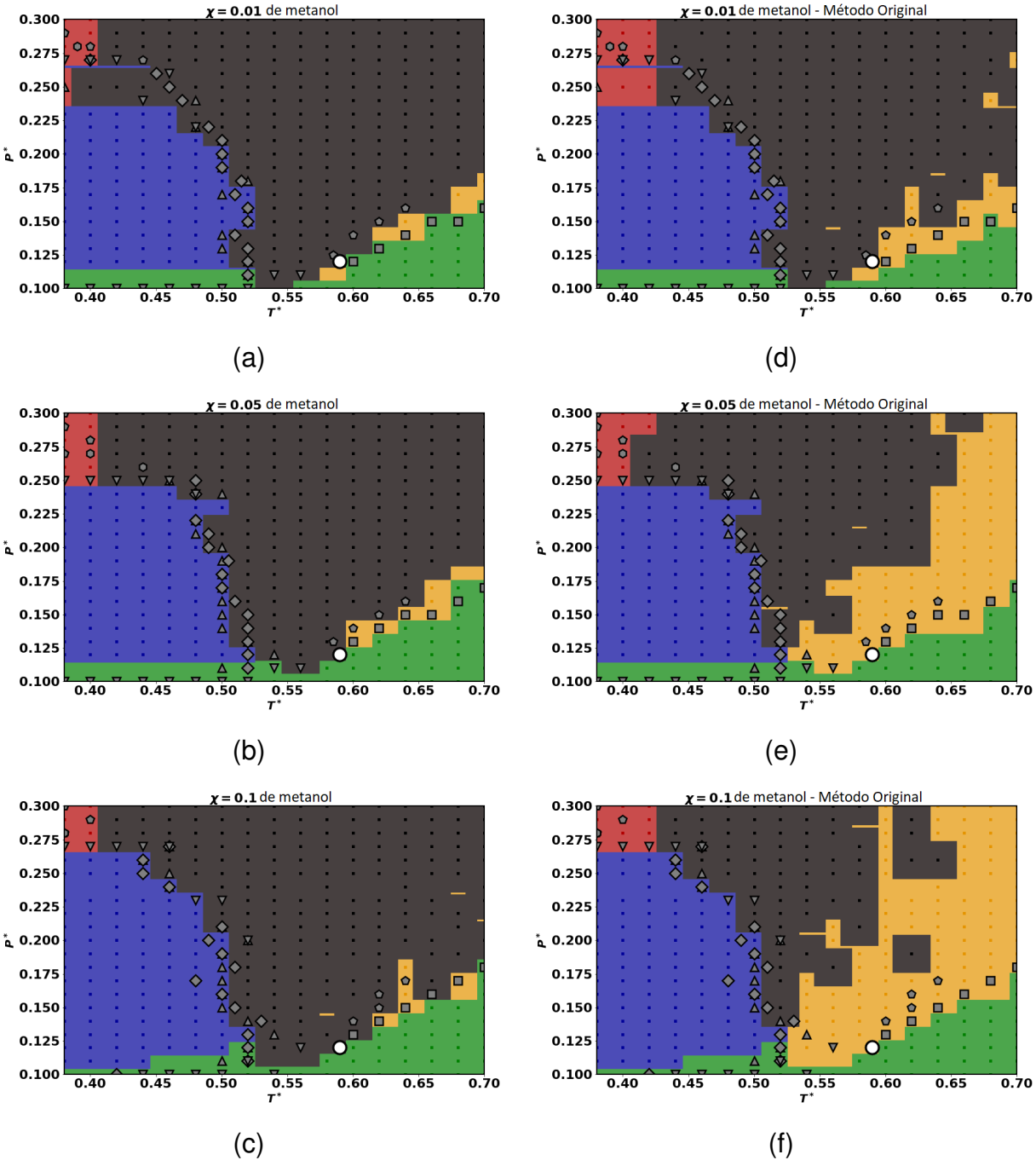


Figura 17 – Diagrama de fase na região $0.38 \leq \mathbf{T}^* \leq 0.70$ e $0.10 \leq \mathbf{P}^* \leq 0.30$, predito pela rede neural usando BOOPs e BOOPs médios para a mistura água-metanol com concentração (a) $\chi = 0.01$, (b) $\chi = 0.05$ e (c) $\chi = 0.1$ e incluindo parâmetros médios-médios para mesma mistura e concentrações (d) $\chi = 0.01$, (e) $\chi = 0.05$ e (f) $\chi = 0.1$. A região com pontos azuis corresponde a configurações classificadas como fase II, a região com pontos vermelhos como fase III, com pontos verdes como fase LDL e amarelo como HDL. A região com pontos pretos é fase metaestável.

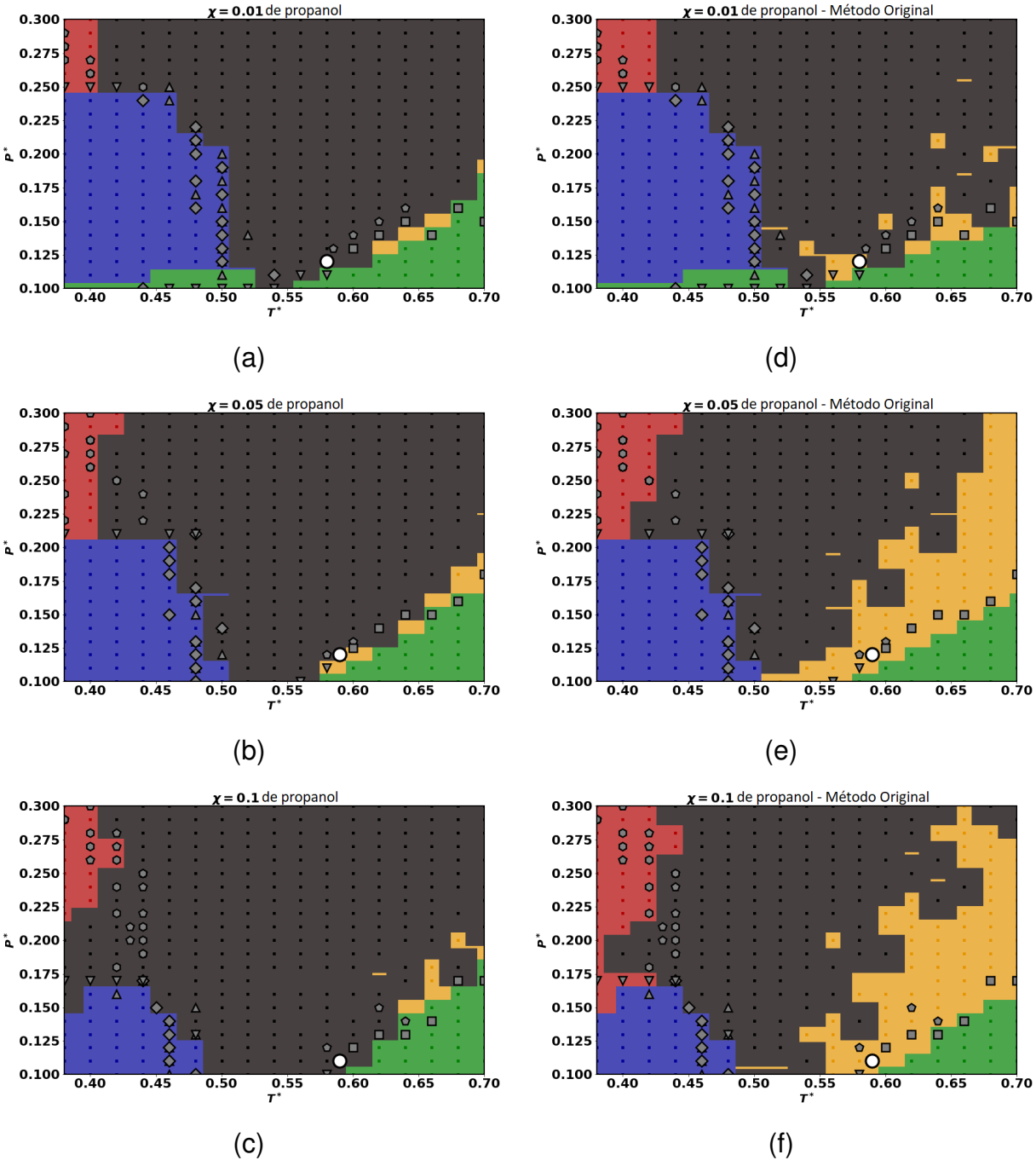


Figura 18 – Diagrama de fase na região $0.38 \leq \mathbf{T}^* \leq 0.70$ e $0.10 \leq \mathbf{P}^* \leq 0.30$, predito pela rede neural usando BOOPs e BOOPs médios para a mistura água-propanol com concentração (a) $\chi = 0.01$, (b) $\chi = 0.05$ e (c) $\chi = 0.1$ e incluindo parâmetros médios-médios para mesma mistura e concentrações (d) $\chi = 0.01$, (e) $\chi = 0.05$ e (f) $\chi = 0.1$. A região com pontos azuis corresponde a configurações classificadas como fase II, a região com pontos vermelhos como fase III, com pontos verdes como fase LDL e amarelo como HDL. A região com pontos pretos é fase metaestável.

Focando agora na região supercrítica, para valores de pressão e temperatura acima do LLCP, na qual não é possível diferenciar de forma precisa duas fases, HDL

e LDL, mas sim uma única fase líquida com duas diferentes regiões: uma com características semelhante à fase líquida de alta densidade, e outra semelhante à fase líquida de baixa densidade. Diferente de uma transição de fase, como no caso da região subcrítica (P e T menores que o LLCP), a separação dessa fase líquida em regiões distintas é dada pela linha de Widom (WL), representada pelos máximos em κ_T, identificados por quadrados cinza nos digramas de fase, figuras 12, 13 e 14, uma extensão da linha de coexistência líquido-líquido na qual se encontra a maior flutuação dos parâmetros de ordem [105, 106]. Percebe-se que a NN classifica de forma efetiva as configurações com fase HDL e LDL, próximas à região de transição entre essas fases e próximas à WL. Novamente, uma análise populacional é implementada para compreender o comportanto das estruturas nessa região do DF, dessa vez realizada para todas as pressões de diferentes isotermas, figura 19, e para todas as temperaturas de diferentes isóbaras, figura 20, para a mistura água-etanol com concentração $\chi = 0.1$.

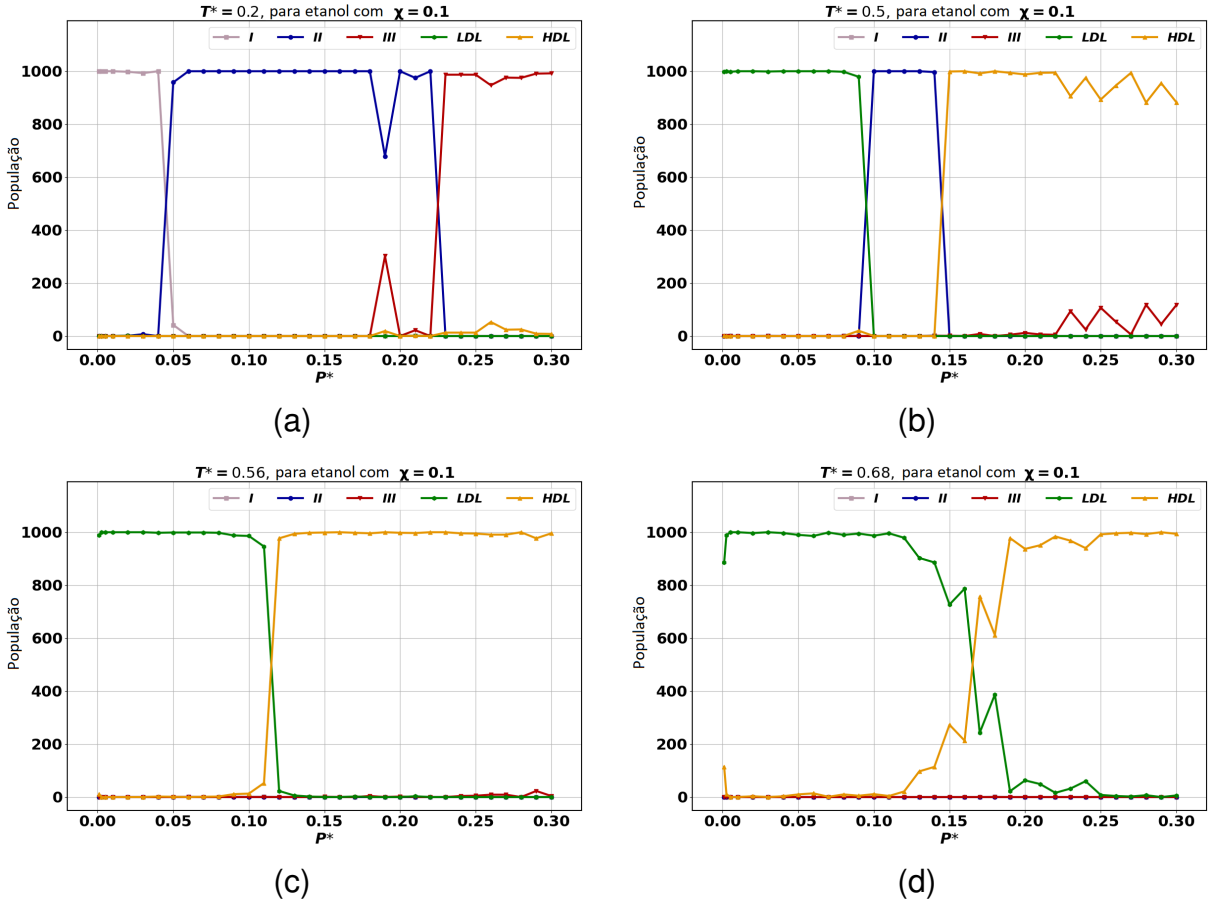


Figura 19 – Populações em função da pressão para a mistura água-etanol com concentração igual a 0.1 e para valores de temperatura igual a (a) 0.2, (b) 0.5, (c) 0.56 e (d) 0.68.

A partir da isoterma $T^* = 0.20$ na figura 19 (a) nota-se que a transição $I - II$ é abrupta, no sentido que praticamente todas as partículas estão com a mesma estru-

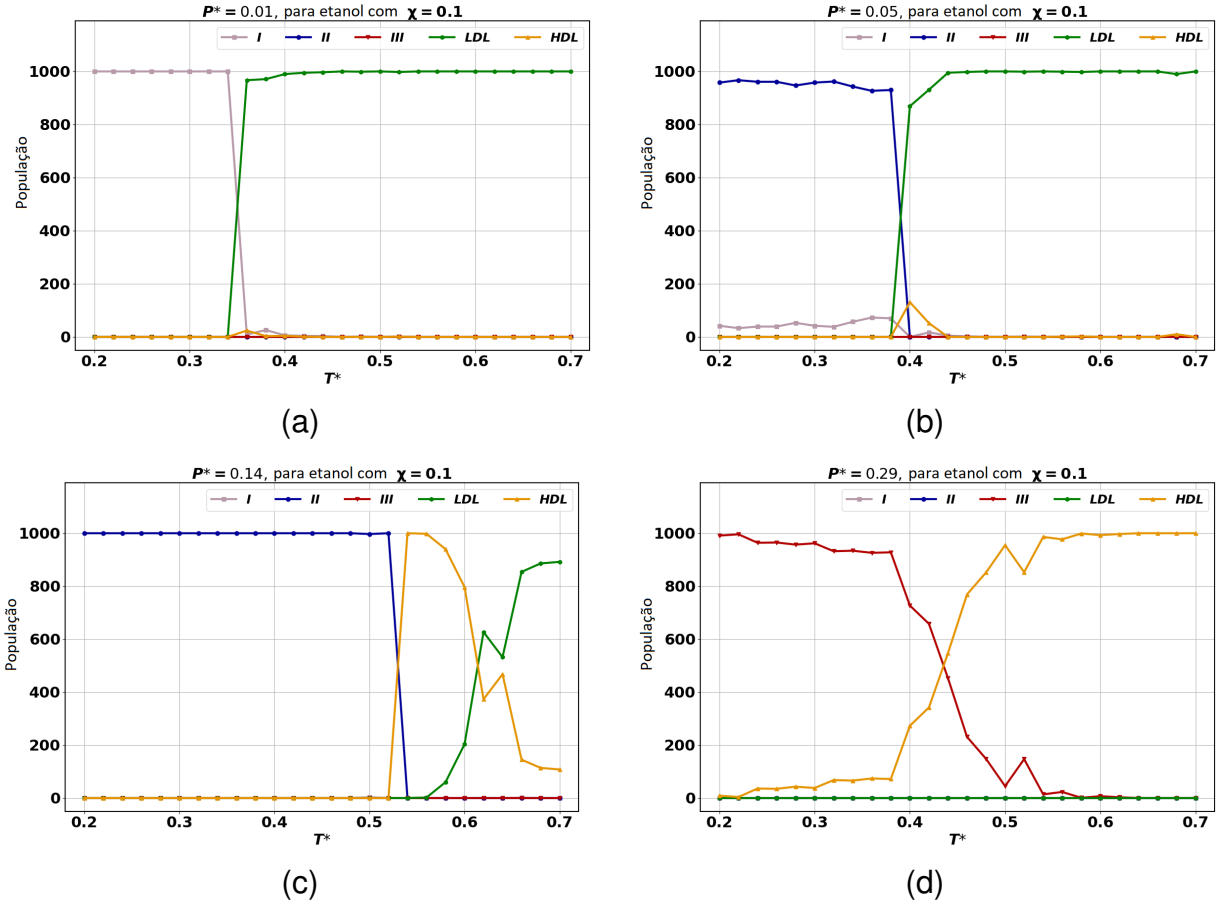


Figura 20 – Populações em função da temperatura para a mistura água-etanol com concentração igual a 0.1 e para valores de pressão igual a (a) 0.01, (b) 0.05, (c) 0.14 and (d) 0.29.

tura BCC e, ao aumentar a pressão, todas as partículas transicionam para a estrutura HCP. Já para a transição $II - III$, observam-se flutuações no número de partículas classificadas como uma ou outra fase, em específico para pressões indo de $P^* = 0.19$ até $P^* = 0.22$, comportamento esperado dada a metaestabilidade presente na fase amorfa. O mesmo comportamento de mudança abrupta de fase faz-se presente tanto no caso da mudança de LDL para II, nas figuras 19 (b) e 19 (c), quanto no caso de II para HDL na figura 19 (b). Nesse último caso ainda é percebida novamente a influência da fase metaestável, onde para pressões mais altas o número de partículas classificadas como amorfas dentro da fase líquida de alta densidade é não nulo. Diferente de todos os casos já descritos, a mudança da fase LDL para HDL para a temperatura supercrítica $T^* = 0.68$ ocorre de forma gradual, com o número de partículas classificadas como líquido de baixa densidade diminuindo aos poucos ao aumentar a temperatura, sendo substituídas por partículas classificadas como líquido de alta densidade. Essa diferença no comportamento da população para a região sub e supercrítica permite definir quando ocorre uma transição de fase $LDL - HDL$ e quando é cruzada a linha de Widom, indo da fase líquida com carcterísticas semelhantes à

fase de baixa densidades para àquela com características semelhantes ao líquido de alta densidade.

Todas essas suposições continuam válidas analisando a população em função da temperatura para diferentes isóbaras. Na figura 20 ocorrem transições abruptas da fase I para LDL, 20 (a), e da fase II para LDL, 20 (b). Na figura 20 (c), nota-se a transição abrupta da fase sólida HCP para o líquido de alta densidade e, para temperaturas mais elevadas, o cruzamento da linha de Widom indo da fase líquida mais parecida com a fase HDL para aquela mais parecida com a fase LDL. Para a transição $III - HDL$, na figura 20 (d), observa-se um comportamento semelhante ao cruzamento da linha de Widom, uma mudança gradual da população, com partículas amorfas dando espaço para partículas HDL conforme a temperatura é elevada, até que a fase metaestável é suprimida e o sistema é uniforme na fase HDL. Diferente da mudança $LDL - HDL$ na região supercrítica, essa variação na população representa sim uma transição de fase. No entanto, diferente das outras transições sólido-sólido, sólido-líquido e líquido-líquido, de primeira ordem, essa é a única transição contínua que ocorre nesse sistema.

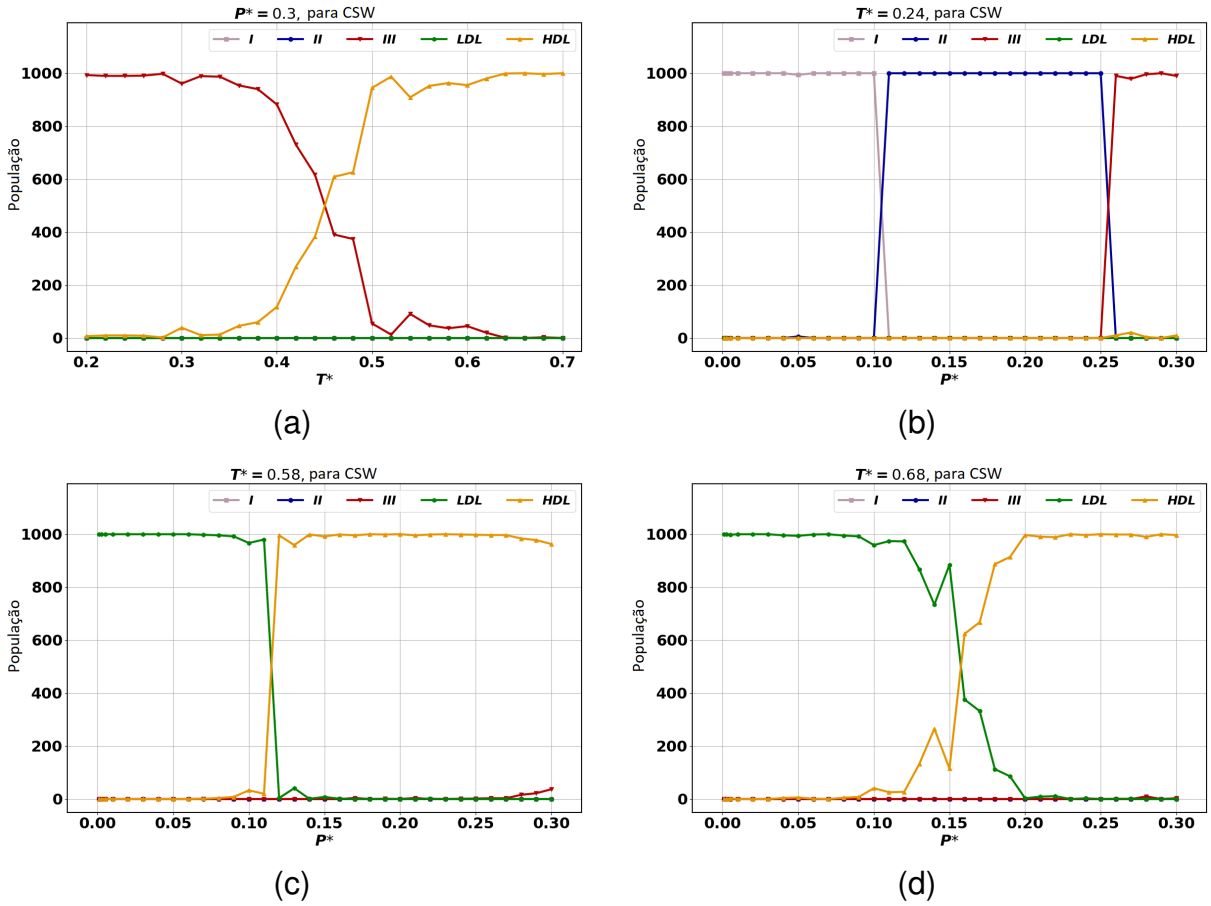


Figura 21 – População para água pura (a) em função da temperatura para pressão igual a 0.3, e como função da pressão para valores de temperatura igual a (b) 0.24, (c) 0.58 e (d) 0.68.

Na figura 21 são expostas populações da água pura para diferentes casos, as quais mostram que as considerações feitas na análise populacional são traduzíveis para o caso sem álcool. Para todas as outras misturas com diferentes concentrações, também foram observados os mesmos comportamentos.

Esses resultados mostram que, além de conseguir classificar com precisão as diferentes fases das misturas água-álcool, a abordagem de aprendizado de máquina possibilita diferenciar quando a transição líquido-líquido ocorre na região supercrítica ou subcrítica, e se as demais transições são contínuas ou descontínuas. Logo, é possível analisar extensivamente as fases do sistema escolhido, sendo necessário somente o cálculo de um conjunto pré-definido de parâmetros. Os resultados foram submetidos recentemente para publicação, e o *preprint* já está disponível no repositório *arxiv* [107].

4 CONCLUSÃO E PERSPECTIVAS

Neste trabalho foi utilizada uma rede neural para classificar as fases de misturas água-álcool, com diferentes álcoois em diferentes concentrações, no regime super resfriado. O estudo desses sistemas surge como uma expansão do trabalho de Marques *et al.* [47], no qual uma mistura água-metanol é estudada, incluindo cadeias de álcoois maiores para verificar o efeito do tamanho do soluto [48]. A partir disso, utilizou-se a abordagem de aprendizado de máquina, inspirada nos estudos de Martelli *et al.* [82] para água super resfriada e de Boattini e colaboradores [91] para classificação de fases de sistemas binários, que se apresenta como uma alternativa para análise termodinâmica realizada em [48].

A abordagem propicia a classifição das fases apresentadas pelas diversas misturas em diferentes concentrações, mostrando grande coerência com os resultados encontrados pela análise termodinâmica. Ainda, se mostra como um método mais rápido e enxuto, necessitando somente de um conjunto pré-definido de parâmetros, evitando o cálculo e a análise de quantidades específicas para cada transição.

Ainda, o modelo desenvolvido adiciona parâmetros médios-médios, que carregam informação estrutural referente à terceiros vizinhos, ao conjunto de parâmetros de ordem orientacional, para realizar o mapeamento da estrutura local dos sistemas em um único vetor. Essa nova implementação possibilita a investigação do comportamento da fase metaestável e como a estrutura local é influenciada pelo ordenamento de curto e longo alcance, e ainda como essas características do sistema são relacionadas à quantidade de álcool.

Por fim, como a rede neural retorna a fase de cada partícula de cada configuração, a partir de uma análise populacional é possível determinar quais transições são contínuas, quais são descontínuais, ou ainda diferenciar quando está ocorrendo uma transição de fase líquido-líquido do cruzamento da linha de Widom, diferenciando a região subcrítica daquela super crítica.

REFERÊNCIAS

[1] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Oxford University Press, Oxford, 1987.

[2] T. Schneider and E. Stoll. Molecular-dynamics study of structural-phase transitions. I. One-component displacement models. *Phys. Rev. B*, 13(3), 1976.

[3] J.S. Rowlinson and F.L. Swinton. *Liquids and Liquid Mixtures*. Butterworths monographs in chemistry and chemical engineering. Butterworth Scientific, 1982.

[4] G. Franzese and M. Rubi. *Aspects of Physical Biology: Biological Water, Protein Solutions, Transport and Replication*. Lecture Notes in Physics. Springer Berlin Heidelberg, 2008.

[5] Y. Koga. *Solution Thermodynamics and its Application to Aqueous Solutions: A Differential Approach*. Elsevier Science, 2007.

[6] E. Ruckenstein and I.L. Shulgin. *Thermodynamics of Solutions: From Gases to Pharmaceutics to Proteins*. SpringerLink: Springer e-Books. Springer New York, 2009.

[7] Niels K.J. Hermkens, Ruud L.E.G. Aspers, Martin C. Feiters, Floris P.J.T. Rutjes, and Marco Tessari. Trace analysis in water-alcohol mixtures by continuous p-H2 hyperpolarization at high magnetic field. *Magnetic Resonance in Chemistry*, 56(7):633–640, 2018.

[8] Thi Vi Na Nguyen, Lydie Paugam, Philippe Rabiller, and Murielle Rabiller-Baudry. Study of transfer of alcohol (methanol, ethanol, isopropanol) during nanofiltration in water/alcohol mixtures. *Journal of Membrane Science*, 601:117907, 2020.

[9] Paulina Prslja, Enrique Lomba, Paula Gómez-Álvarez, Tomaz Urbic, and Eva G. Noya. Adsorption of water, methanol, and their mixtures in slit graphite pores. *The Journal of Chemical Physics*, 150(2):024705, 2019.

[10] Leland M Vane. Review: membrane materials for the removal of water from industrial solvents by pervaporation and vapor permeation. *Journal of Chemical Technology & Biotechnology*, 94(2):343–365, 2019.

[11] F. Franks, Royal Society of Chemistry (Great Britain), and Royal Society of Chemistry (Great Britain). *Water: A Matrix of Life*. RSC paperbacks. Royal Society of Chemistry, 2000.

[12] Noel T. Southall, Ken A. Dill, and A. D. J. Haymet. A view of the hydrophobic effect. *The Journal of Physical Chemistry B*, 106(3):521–533, 2002.

[13] Erte Xi and Amish J. Patel. The hydrophobic effect, and fluctuations: The long and the short of it. *Proceedings of the National Academy of Sciences*, 113(17):4549–4551, 2016.

[14] M. Chaplin. Anomalous properties of water. `http://www.lsbu.ac.uk/water/anmlies.html`, July 2020.

[15] R. Podgornik. Water and life: the unique properties of H2O. *Journal of Biological Physics*, 37:163–165, 2011.

[16] Johannes Bachler, Philip H. Handle, Nicolas Giovambattista, and Thomas Loerting. Glass polymorphism and liquid–liquid phase transition in aqueous solutions: experiments and computer simulations. *Phys. Chem. Chem. Phys.*, 21:23238–23268, 2019.

[17] Pierre Lucas, Shuai Wei, and C. Austen Angell. Liquid-liquid phase transitions in glass-forming systems and their implications for memory technology. *International Journal of Applied Glass Science*, 11(2):236–244, 2020.

[18] Paola Gallo, Katrin Amann-Winkel, Charles Austen Angell, Mikhail Alexeevich Anisimov, Frédéric Caupin, Charusita Chakravarty, Erik Lascaris, Thomas Loerting, Athanassios Zois Panagiotopoulos, John Russo, Jonas Alexander Sellberg, Harry Eugene Stanley, Hajime Tanaka, Carlos Vega, Limei Xu, and Lars Gunnar Moody Pettersson. Water: A tale of two liquids. *Chemical Reviews*, 116(13):7463–7500, 2016. PMID: 27380438.

[19] C. Austen Angell. Two phases? *Nat. Mater.*, 13:673–675, 2014.

[20] Peter Poole, Francesco Sciortino, Ulrich Essmann, and H. Stanley. Phase-behavior of metastable water. *Nature*, 360:324–328, 11 1992.

[21] Peter H. Poole, Richard K. Bowles, Ivan Saika-Voivod, and Francesco Sciortino. Free energy surface of ST2 water near the liquid-liquid phase transition. *The Journal of Chemical Physics*, 138(3):034505, 2013.

[22] David T. Limmer and David Chandler. The putative liquid-liquid transition is a liquid-solid transition in atomistic models of water. *The Journal of Chemical Physics*, 135(13):134503, 2011.

[23] David T. Limmer and David Chandler. The putative liquid-liquid transition is a liquid-solid transition in atomistic models of water. II. *The Journal of Chemical Physics*, 138(21):214504, 2013.

[24] Jeremy C. Palmer, Roberto Car, and Pablo G. Debenedetti. The liquid–liquid transition in supercooled ST2 water: a comparison between umbrella sampling and well-tempered metadynamics. *Faraday Discuss.*, 167:77–94, 2013.

[25] Jeremy C. Palmer, Peter H. Poole, Francesco Sciortino, and Pablo G. Debenedetti. Advances in computational studies of the liquid–liquid transition in water and water-like models. *Chemical Reviews*, 118(18):9129–9151, 2018. PMID: 30152693.

[26] H.E. Stanley, L. Cruz, S.T. Harrington, P.H. Poole, S. Sastry, F. Sciortino, F.W. Starr, and R. Zhang. Cooperative molecular motions in water: The liquid-liquid critical point hypothesis. *Physica A: Statistical Mechanics and its Applications*, 236(1):19 – 37, 1997. Proceedings of the Workshop on Current Problems in Complex Fluids.

[27] Osamu Mishima and H. Stanley. The relationship between liquid, supercooled and glassy water. *Nature*, 396, 11 1998.

[28] H. Stanley, Sergey Buldyrev, Osamu Mishima, M Sadr-Lahijany, Antonio Scala, and Francis Starr. Unsolved mysteries of water in its liquid and glassy phases. *Journal of Physics: Condensed Matter*, 12:A403, 02 2000.

[29] Francesco Sciortino, Emilia la Nave, and Piero Tartaglia. Physics of the liquid-liquid critical point. *Physical review letters*, 91:155701, 11 2003.

[30] Pablo G Debenedetti. Supercooled and glassy water. *Journal of Physics: Condensed Matter*, 15(45):R1669–R1726, oct 2003.

[31] Philip H. Handle, Thomas Loerting, and Francesco Sciortino. Supercooled and glassy water: Metastable liquid(s), amorphous solid(s), and a no-man's land. *Proceedings of the National Academy of Sciences*, 114(51):13336–13344, 2017.

[32] Nicholas J. Hestand and J. L. Skinner. Perspective: Crossing the Widom line in no man's land: Experiments, simulations, and the location of the liquid-liquid critical point in supercooled water. *The Journal of Chemical Physics*, 149(14):140901, 2018.

[33] Christoph G. Salzmann. Advances in the experimental exploration of water's phase diagram. *The Journal of Chemical Physics*, 150(6):060901, 2019.

[34] Kyung Hwan Kim, Katrin Amann-Winkel, Nicolas Giovambattista, Alexander Späh, Fivos Perakis, Harshad Pathak, Marjorie Ladd Parada, Cheolhee Yang, Daniel Mariedahl, Tobias Eklund, Thomas. J. Lane, Seonju You, Sangmin Jeong, Matthew Weston, Jae Hyuk Lee, Intae Eom, Minseok Kim, Jaeku Park, Sae Hwan Chun, Peter H. Poole, and Anders Nilsson. Experimental observation of the liquid-liquid transition in bulk supercooled water under pressure. *Science*, 370(6519):978–982, 2020.

[35] Pablo G. Debenedetti, Francesco Sciortino, and Gül H. Zerze. Second critical point in two realistic models of water. *Science*, 369(6501):289–292, 2020.

[36] E. A. Jagla. Phase behavior of a system of particles with core collapse. *Phys. Rev. E*, 58:1478–1486, 1998.

[37] E. A. Jagla. Core-softened potentials and the anomalous properties of water. *The Journal of Chemical Physics*, 111(19):8980–8986, 1999.

[38] Alan Oliveira, Paulo Netz, and Marcia Barbosa. Which mechanism underlies the water-like anomalies in core-softened potentials? *The European Physical Journal B*, 64:481–486, 01 2008.

[39] Martin Meyer and H. Eugene Stanley. Liquid–liquid phase transition in confined water: A monte carlo study. *The Journal of Physical Chemistry B*, 103(44):9728–9730, 1999.

[40] Leandro B. Krott, José Rafael Bordin, Ney M. Barraz, and Marcia C. Barbosa. Effects of confinement on anomalies and phase transitions of core-softened fluids. *The Journal of Chemical Physics*, 142(13):134502, 2015.

[41] Yu. D. Fomin, E. N. Tsiok, and V. N. Ryzhov. Inversion of sequence of diffusion and density anomalies in core-softened systems. *J. Chem. Phys.*, 135:234502, 2011.

[42] Pol Vilaseca and Giancarlo Franzese. Isotropic soft-core potentials with two characteristic length scales and anomalous behaviour. *Journal of Non-Crystalline Solids*, 357:419–426, 01 2011.

[43] Giancarlo Franzese. Differences between discontinuous and continuous soft-core attractive potentials: The appearance of density anomaly. *Journal of Molecular Liquids*, 136(3):267 – 273, 2007. EMLG/JMLG 2006.

[44] Pol Vilaseca and Giancarlo Franzese. Softness dependence of the anomalies for the continuous shouldered well potential. *The Journal of Chemical Physics*, 133(8):084507, 2010.

[45] Matej Hus and Tomaz Urbic. Existence of a liquid-liquid phase transition in methanol. *Phys. Rev. E*, 90:062306, 12 2014.

[46] Alan Barros de Oliveira, Giancarlo Franzese, Paulo A. Netz, and Marcia C. Barbosa. Waterlike hierarchy of anomalies in a continuous spherical shouldered potential. *The Journal of Chemical Physics*, 128(6):064901, 2008.

[47] Murilo Sodré Marques, Vinicius Fonseca Hernandes, Enrique Lomba, and José Rafael Bordin. Competing interactions near the liquid-liquid phase transition of core-softened water/methanol mixtures. *Journal of Molecular Liquids*, 320:114420, 2020.

[48] Murilo S. Marques, Vinicius F. Hernandes, and Jose Rafael Bordin. Core-softened water-alcohol mixtures: the solute-size effects. *arXiv:2102.09485*.

[49] Matej Huš, Gianmarco Munaò, and Tomaz Urbic. Properties of a soft-core model of methanol: An integral equation theory and computer simulation study. *The Journal of Chemical Physics*, 141(16):164505, 2014.

[50] Matej Huš, Gašper Žakelj, and Tomaž Urbič. Properties of methanol-water mixtures in a coarse-grained model. *Acta Chimica Slovenica*, 62(3):524–530, 2015.

[51] Gianmarco Munaò and Tomaz Urbic. Structure and thermodynamics of core-softened models for alcohols. *The Journal of Chemical Physics*, 142(21):214508, 2015.

[52] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[53] B. Widrow and M. E. Hoff. Adaptive switching circuits. *IRE WESCON Convention Record*, 1960.

[54] T. M. Cover and P. E. Hart. Nearest Neighbor Pattern Classification. *IEEE Trans. on Information Theory*, 13, 1967.

[55] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceed. Nat. Acad. Science*, 79, 1982.

[56] Y. Abu-Mostafa and J. St. Jacques. Information capacity of the Hopfield model. *IEEE Trans. on Information Theory*, 31, 1985.

[57] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[58] A. Senior, Richard Evans, J. Jumper, J. Kirkpatrick, L. Sifre, Tim Green, Chongli Qin, Augustin Zídek, Alexander W. R. Nelson, A. Bridgland, Hugo Penedones, Stig Petersen, K. Simonyan, Steve Crossan, P. Kohli, David T. Jones, D. Silver, K. Kavukcuoglu, and D. Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577:706–710, 2020.

[59] J. Hermann, Zeno Schätzle, and F. Noé. Deep-neural-network solution of the electronic schrödinger equation. *Nature Chemistry*, pages 1–7, 2020.

[60] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G.R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab. A high-bias, low-variance introduction to machine learning for physicists. *Physics Reports*, 810:1–124, 2019. A high-bias, low-variance introduction to Machine Learning for physicists.

[61] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[62] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[63] Alexander Amini, A. Soleimany, S. Karaman, and D. Rus. Spatial uncertainty sampling for end-to-end control. *ArXiv*, abs/1805.04829, 2018.

[64] Alexander Amini, Ava Soleimany, Sertac Karaman, and Daniela Rus. Spatial uncertainty sampling for end-to-end control. *arXiv:1805.04829*.

[65] Dan Guest, Kyle Cranmer, and Daniel Whiteson. Deep Learning and Its Application to LHC Physics. *Annual Review of Nuclear and Particle Science*, 68(1):161–181, 2018.

[66] Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. Machine learning for molecular simulation. *Annual Review of Physical Chemistry*, 71(1):361–390, 2020.

[67] Matti Hellström and Jörg Behler. *High-Dimensional Neural Network Potentials for Atomistic Simulations*, chapter 3, pages 49–59.

[68] Ying Li, Hui Li, Frank C. Pickard, Badri Narayanan, Fatih G. Sen, Maria K. Y. Chan, Subramanian K. R. S. Sankaranarayanan, Bernard R. Brooks, and Benoît Roux. Machine Learning Force Field Parameters from Ab Initio Data. *Journal of Chemical Theory and Computation*, 13(9):4492–4503, 2017. PMID: 28800233.

[69] V. Botu, R. Batra, J. Chapman, and R. Ramprasad. Machine learning force fields: Construction, validation, and outlook. *The Journal of Physical Chemistry C*, 121(1):511–522, 2017.

[70] James L. McDonagh, Ardita Shkurti, David J. Bray, Richard L. Anderson, and Edward O. Pyzer-Knapp. Utilizing machine learning for efficient parameterization of coarse grained molecular force fields. *Journal of Chemical Information and Modeling*, 59(10):4278–4288, 2019. PMID: 31549507.

[71] R. Durrer, B. Kratochwil, J.V. Koski, A.J. Landig, C. Reichl, W. Wegscheider, T. Ihn, and E. Greplova. Automated tuning of double quantum dots into specific charge states using neural networks. *Phys. Rev. Applied*, 13:054019, 2020.

[72] Justyna P. Zwolak, Thomas McJunkin, Sandesh S. Kalantre, J.P. Dodson, E.R. MacQuarrie, D.E. Savage, M.G. Lagally, S.N. Coppersmith, Mark A. Eriksson, and Jacob M. Taylor. Autotuning of double-dot devices in situ with machine learning. *Phys. Rev. Applied*, 13:034075, 2020.

[73] Andrew W. Long and Andrew L. Ferguson. Nonlinear machine learning of patchy colloid self-assembly pathways and mechanisms. *The Journal of Physical Chemistry B*, 118(15):4228–4244, 2014. PMID: 24660984.

[74] Wesley F. Reinhart, Andrew W. Long, Michael P. Howard, Andrew L. Ferguson, and Athanassios Z. Panagiotopoulos. Machine learning for autonomous crystal structure identification. *Soft Matter*, 13:4733–4745, 2017.

[75] Xiaochuan Zhao, Chenyi Liao, Yong-Tao Ma, Jonathon B. Ferrell, Severin T. Schneebeli, and Jianing Li. Top-down Multiscale Approach To Simulate Peptide Self-Assembly from Monomers. *Journal of Chemical Theory and Computation*, 15(3):1514–1522, 2019. PMID: 30677300.

[76] Carl S. Adorf, Timothy C. Moore, Yannah J. U. Melle, and Sharon C. Glotzer. Analysis of self-assembly pathways with unsupervised machine learning algorithms. *The Journal of Physical Chemistry B*, 124(1):69–78, 2020. PMID: 31813215.

[77] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91:045002, Dec 2019.

[78] Eliska Greplova, Agnes Valenti, Gregor Boschung, Frank Schäfer, Niels Lörch, and Sebastian D Huber. Unsupervised identification of topological phase transitions using predictive models. *New Journal of Physics*, 22(4):045003, 2020.

[79] Emanuele Boattini, Marjolein Dijkstra, and Laura Filion. Unsupervised learning for local structure detection in colloidal systems. *The Journal of Chemical Physics*, 151(15):154901, 2019.

[80] Juan Carrasquilla and Roger G Melko. Machine learning phases of matter. *Nature Physics*, 13(5):431–434, 2017.

[81] Takamichi Terao. A machine learning approach to analyze the structural formation of soft matter via image recognition. *Soft Materials*, 18(0):215–227, 2020.

[82] Fausto Martelli, Fabio Leoni, Francesco Sciortino, and John Russo. Connection between liquid and non-crystalline solid phases in water. *The Journal of Chemical Physics*, 153(10):104503, 2020.

[83] Philipp Geiger and Christoph Dellago. Neural networks for local structure detection in polymorphic systems. *The Journal of Chemical Physics*, 139(16):164105, 2013.

[84] Thomas E. Gartner, Linfeng Zhang, Pablo M. Piaggi, Roberto Car, Athanassios Z. Panagiotopoulos, and Pablo G. Debenedetti. Signatures of a liquid–liquid transition in an ab initio deep neural network model for water. *Proceedings of the National Academy of Sciences*, 117(42):26040–26046, 2020.

[85] B. Monserrat, J.G. Brandenburg, E.A. Engel, and B. Cheng. Liquid water contains the building blocks of diverse ice phases. *Nature Communications*, 11:5757, 2020.

[86] Maxwell Fulford, Matteo Salvalaglio, and Carla Molteni. DeepIce: A Deep Neural Network Approach To Identify Ice and Water Molecules, journal = Journal of Chemical Information and Modeling. 59(5):2141–2149, 2019. PMID: 30875217.

[87] Samuel S. Schoenholz. Combining machine learning and physics to understand glassy systems. *Journal of Physics: Conference Series*, 1036:012021, jun 2018.

[88] Emanuele Boattini, Susana Marín-Aguilar, Saheli Mitra, Giuseppe Foffi, Frank Smallenburg, and Laura Filion. Autonomously revealing hidden local structures in supercooled liquids. *Nature Communications*, 11:5479, 2020.

[89] Paul J. Steinhardt, David R. Nelson, and Marco Ronchetti. Bond-orientational order in liquids and glasses. *Phys. Rev. B*, 28:784–805, Jul 1983.

[90] Wolfgang Lechner and Christoph Dellago. Accurate determination of crystal structures based on averaged local bond order parameters. *The Journal of Chemical Physics*, 129(11):114707, 2008.

[91] Emanuele Boattini, Michel Ram, Frank Smallenburg, and Laura Filion. Neural-network-based order parameters for classification of binary hard-sphere crystal structures. *Molecular Physics*, 116(21-22):3066–3075, 2018.

[92] L. Verlet. Computer "Experiments"on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Physical Review*, 159, 1967.

[93] José Bordin and Marcia Barbosa. Flow and structure of fluids in functionalized nanopores. *Physica A: Statistical Mechanics and its Applications*, 467, 10 2016.

[94] Axel Arnold, Olaf Lenz, Stefan Kesselheim, Rudolf Weeber, Florian Fahrenberger, Dominic Roehm, Peter Košovan, and Christian Holm. Espresso 3.1: Molecular dynamics software for coarse-grained models. In Michael Griebel and Marc Alexander Schweitzer, editors, *Meshfree Methods for Partial Differential Equations VI*, pages 1–23, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[95] F. Weik et al. ESPResSo 4.0 – an extensible software package for simulating soft matter systems. *The European Physical Journal Special Topics*, 227, 2019.

[96] A. Kolb and B. Dünweg. Optimized constant pressure stochastic dynamics. *The Journal of Chemical Physics*, 111(10):4453–4459, 1999.

[97] Alexander Stukowski. Visualization and analysis of atomistic simulation data with OVITO-the Open Visualization Tool. *MODELLING AND SIMULATION IN MATERIALS SCIENCE AND ENGINEERING*, 18(1), JAN 2010.

[98] Vyas Ramasubramani, Bradley D. Dice, Eric S. Harper, Matthew P. Spellings, Joshua A. Anderson, and Sharon C. Glotzer. freud: A software suite for high throughput analysis of particle simulation data. *Computer Physics Communications*, 254:107275, 2020.

[99] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[100] Chris H. Rycroft. VORO++: A three-dimensional Voronoi cell library in C++. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 19(4):041111, 2009.

[101] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors,

Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. JMLR Workshop and Conference Proceedings.

[102] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic gradient descent. In *ICLR: International Conference on Learning Representations*, pages 1–15, 2015.

[103] François Chollet et al. Keras. `https://keras.io`, 2015.

[104] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[105] V. Holten, C. E. Bertrand, M. A. Anisimov, and J. V. Sengers. Thermodynamics of supercooled water. *The Journal of Chemical Physics*, 136(9):094507, 2012.

[106] Valentino Bianco and Giancarlo Franzese. Hydrogen bond correlated percolation in a supercooled water monolayer as a hallmark of the critical region. *Journal of Molecular Liquids*, 285:727–739, 2019.

[107] Vinicius F. Hernandes, Murilo S. Marques, and José R. Bordin. Phase classification using neural networks: application to supercooled, polymorphic core-softened mixtures. *arXiv:2106.13189*.

5 APÊNDICES

5.1 Apêndice A: algoritmos desenvolvidos para obtenção dos resultados

A seguir é exposto o algoritmo em python utilizado para o cálculo do conjunto de dados do vetor $\mathbf{q}$, para uma configuração específica

```
# pacotes utilizados no codigo sao importados
import numpy as np
import freud as fd
import glob
import os
import pandas as pd
from statistics import mean

#caminho para a pasta onde estao os snapshots
path_snaps = '/home/vini/Documents/pesquisa/murilo/snaps/'
#caminho para onde se quer salvas os BOOPs
path_out_qs = '/home/vini/Documents/pesquisa/murilo/new_qs_third_shell/'
# nome da pasta da mistura (ou agua pura)
alcohol = 'pure-water/'
# nome da pasta com concentracao da mistura (vazio no caso de agua pura)
frac = ''#'0.05/'

# usando o pacote glob, sao selecionados, um a um todos os
# arquivos com terminacao .xyz presentes em uma determina pasta
for file in glob.glob(path_snaps + alcohol + frac + '*.xyz'):
  # usando o pacote os, verifica-se o tamanho do arquivo
  filesize = os.stat(file).st_size
  # caso o arquivo tenha tamanho zero (arquivo vazia),
  # passa para o proximo arquivo
  if(filesize == 0):
    continue

  # dependendo de como os nomes dos arquivos de entrada estao organizados
  # trechos especificos dos nomes dos arquivos sao salvos nas
```

```
30  # variaveis abaixos para encontrar de forma autonoma arquivos
31  # com determinada pressao e temperatura
32  startP = 'P-'
33  endP = '-T'
34  startT = 'T-'
35  endT = '.xyz'
36  # valor de pressao e temperatura (string) do arquivo para o qual os
37  # BOOPs serao calculados
38  P = file[file.find(startP)+len(startP):file.find(endP)]
39  T = file[file.find(startT)+len(startT):file.find(endT)]
40
41  # um arquivo de saida, que vai conter o conjunto de BOOPs para
     determinados
42  # valores de P e T, e criado
43  file_out_qs = open(path_out_qs + alcohol + frac + 'qs_ws_data_P{}_T{}_.
     dat'.format(P,T), 'w+')
44
45  # lendo a primeira linha do arquivo de entrada, encontra-se o numero de
     particulas do sistema
46  # isso e valido para arquivos formatados em .xyz, formato de padrao para
     leitura do software Ovito
47  N = int(np.loadtxt(file, usecols=0, max_rows=1))
48  # usando o metodo genfromtxt, o numero total de linhas do arquivo e salvo
      em uma variavel
49  number_of_lines = int(len(np.genfromtxt(file, usecols=0, dtype=str)))
50
51  # sabendo o numero de particulas do sistema e o numero de linhas, usando
     a biblioteca pandas
52  # sao importados todos os dados, somente para o ultimo timestep do
     snapshot
53  data = pd.read_csv(file, sep = '\s+', skiprows= number_of_lines - N,
     header = None)
54  # entao e usada a condicao abaixo para verificar se algum valor esta
     faltando
55  # caso isso aconteca, imprimi-se uma mensagem e passa-se para o proximo
     arquivo
56  if(data.isnull().values.any()):
57    print('The file for P = {} and T = {} has some error! Check it!'.format
     (P,T))
58    continue
59
60  # as posicoes de todas as particulas, independente do tipo de molecula
61  # ou radicando (no caso do alcool), sao armazenadas em uma variavel
62  positions_all = np.loadtxt(file, usecols = (1,2,3), skiprows=
     number_of_lines - N, max_rows=N)
63  # os tipos de particula, correspondente a letra na primeira coluna do
     snapshot, sao armazenados
```

```
64  # para esse caso especifico, temos
65  # N --> agua
66  # O --> OH
67  # F --> CH3
68  # S --> CH2
69  molecule = np.genfromtxt(file, dtype=str, skip_header= number_of_lines -
     N, max_rows=N, usecols=0)
70
71  # o vetor de posicoes e divididos em vetores unicos para posicao x, y e z
72  x, y, z = positions_all[:, 0], positions_all[:, 1], positions_all[:, 2]
73  # a partir dos maximos e minimos de x, y e z, encontra-se o tamanho da
     caixa de simulacao
74  Lx, Ly, Lz = x.max()-x.min(), y.max()-y.min(), z.max()-z.min()
75
76  # usando o pacote freud, definise uma caixa de tamanho igual da caixa de
     simulacao
77  box = fd.box.Box(Lx, Ly, Lz)
78  vor = fd.locality.Voronoi() # objeto da classe Voronoi e criado
79  # listas vazias para depois armazenas posicoes de particulas do tipo N e
     O,
80  # somente N e somente O sao criadas
81  positions = []
82  positions_N = []
83  positions_O = []
84
85  # varrendo o tipo de particulas sao selecionadas somente aquelas do tipo
     N ou O,
86  # e armazenadas nas listas correspondentes
87  for i in range(len(positions_all)):
88    if(molecule[i] != 'F' and molecule[i] != 'S'):
89      positions.append(positions_all[i])
90      if(molecule[i] == 'N'):
91        positions_N.append(positions_all[i])
92      else:
93        positions_O.append(positions_all[i])
94
95
96  # listas vazias para os BOOPs sao criadas
97  qs, ws, qs_avg, ws_avg, qs_avg_avg, ws_avg_avg = [], [], [], [], [], []
98  qs_O, ws_O, qs_avg_O, ws_avg_O, qs_avg_avg_O, ws_avg_avg_O = [], [], [],
     [], [], []
99  qs_N, ws_N, qs_avg_N, ws_avg_N, qs_avg_avg_N, ws_avg_avg_N = [], [], [],
     [], [], []
100
101 for l in range(3,13):  # varrem-se valores de l indo de 3 a 12
102
```

```
103     # ----- calculo dos BOOPs para todas as particulas, independente da
        ligacao ----
104     ngbs = vor.compute((box, positions)).nlist   # relacao de vizinhanca e
        armazenada
105     # um objeto q da classe Steinhardt, para um l especifico, e criado
106     q = fd.order.Steinhardt(l = l)
107     # usando q, para a caixa de simulacao box,
108     # sao calculados os ql's para todas as particulas, com relacao de
        vizinhanca dada
109     # por ngbs, independente da ligacao
110     q_part = q.compute((box,positions), ngbs).particle_order
111     # os valores de ql de todas as particulas sao adicionadas a lista qs
112     qs.append(q_part)
113
114     # o mesmo processo e repetido para os parametros medios, definindo
        average=True
115     # na classe Steinhardt
116     qavg = fd.order.Steinhardt(l = l, average = True)
117     qavg_part = qavg.compute((box,positions), ngbs).particle_order
118     qs_avg.append(qavg_part)
119
120     # e criada uma lista que contem a identificao (id = ordem no vetor
        positions)
121     #  de cada particula, acompanhada dos ids de todos seus vizinhos
122     ngbs_indices_list = [[i] for i in range(len(positions))]
123     for i, j in ngbs[:]:
124       if(i == ngbs_indices_list[i][0]):
125         ngbs_indices_list[i].append(j)
126
127     # e criado um vetor que contem a media dos q-medios para a relacao de
        vizinhanca
128     # contida na lista criada acima, para todas as particulas do sistema.
129     # entao, esses valores sao adicionados a lista qavg_avg_part
130     avg_aux = [[] for i in range(len(positions))]
131     qavg_avg_part = [0 for i in range(len(positions))]
132     for i in range(len(ngbs_indices_list)):
133       for j in range(len(ngbs_indices_list[i])):
134         avg_aux[i].append(qavg_part[ngbs_indices_list[i][j]])
135       qavg_avg_part[i] = mean(avg_aux[i])
136     qs_avg_avg.append(qavg_avg_part)
137
138
139     # processo semelhante ao realizado acima e feito somente para os
        valores pares de l
140     # para os BOOPs cubicos, definindo wl=True na funcao Steinhardt
141     if(int(l) % 2 == 0):
142       w = fd.order.Steinhardt(l = l, wl = True)
```

```
143       w_part = w.compute((box,positions), ngbs).particle_order
144       ws.append(w_part)
145       wavg = fd.order.Steinhardt(l = l, wl = True, average = True)
146       wavg_part = wavg.compute((box,positions), ngbs).particle_order
147       ws_avg.append(wavg_part)
148
149       avg_aux = [[] for i in range(len(positions))]
150       wavg_avg_part = [0 for i in range(len(positions))]
151
152       for i in range(len(ngbs_indices_list)):
153         for j in range(len(ngbs_indices_list[i])):
154           avg_aux[i].append(wavg_part[ngbs_indices_list[i][j]])
155         wavg_avg_part[i] = mean(avg_aux[i])
156
157       ws_avg_avg.append(wavg_avg_part)
158
159
160     # um vetor de zeros com o tamanho da lista positions e criado, para ser
      usado
161     # como valores para os BOOPs referentes as particulas de alcool, quando
      somente
162     # ligacoes particulas de agua existem no sistema, OU para
163     # particulas de agua, quando somente particulas de alcool existem
164     aux_0 = np.zeros((len(positions),))
165
166     #  ----- calculo dos BOOPs considerando somente interacoes N-N ----
167     # o processo todo e semelhante ao calculos dos BOOPs para todas
    particulas,
168     # independente do tipo de ligacao. A diferenca e que sao considerados
    somente as
169     # posicoes e a relacao de vizinhanca entre particulas de agua (N)
170     # ---------
171     # caso o sistema seja composto por somente alcool, o vetor de zeros
    para definir os
172     # BOOPs das ligacoes N-N
173     if(len(positions_N) == 0):
174       qs_N.append(aux_0)
175       qs_avg_N.append(aux_0)
176       qs_avg_avg_N.append(aux_0)
177       if(int(l) % 2 == 0):
178         ws_N.append(aux_0)
179         ws_avg_N.append(aux_0)
180         ws_avg_avg_N.append(aux_0)
181     # caso contratio, o calculo e realizado normalmente
182     else:
183       ngbs = vor.compute((box, positions_N)).nlist
184       q = fd.order.Steinhardt(l = l)
```

```
185       q_part = q.compute((box,positions_N), ngbs).particle_order
186       qs_N.append(q_part)
187       qavg = fd.order.Steinhardt(l = l, average = True)
188       qavg_part = qavg.compute((box,positions_N), ngbs).particle_order
189       qs_avg_N.append(qavg_part)
190
191       ngbs_indices_list = [[i] for i in range(len(positions_N))]
192       for i, j in ngbs[:]:
193         if(i == ngbs_indices_list[i][0]):
194           ngbs_indices_list[i].append(j)
195
196       avg_aux = [[] for i in range(len(positions_N))]
197       qavg_avg_part = [0 for i in range(len(positions_N))]
198
199       for i in range(len(ngbs_indices_list)):
200         for j in range(len(ngbs_indices_list[i])):
201           avg_aux[i].append(qavg_part[ngbs_indices_list[i][j]])
202         qavg_avg_part[i] = mean(avg_aux[i])
203
204       qs_avg_avg_N.append(qavg_avg_part)
205
206       if(int(l) % 2 == 0):
207         w = fd.order.Steinhardt(l = l, wl = True)
208         w_part = w.compute((box,positions_N), ngbs).particle_order
209         ws_N.append(w_part)
210         wavg = fd.order.Steinhardt(l = l, wl = True, average = True)
211         wavg_part = wavg.compute((box,positions_N), ngbs).particle_order
212         ws_avg_N.append(wavg_part)
213
214         avg_aux = [[] for i in range(len(positions_N))]
215         wavg_avg_part = [0 for i in range(len(positions_N))]
216
217         for i in range(len(ngbs_indices_list)):
218           for j in range(len(ngbs_indices_list[i])):
219             avg_aux[i].append(wavg_part[ngbs_indices_list[i][j]])
220           wavg_avg_part[i] = mean(avg_aux[i])
221
222         ws_avg_avg_N.append(wavg_avg_part)
223
224     #  ----- calculo dos BOOPs considerando somente interacoes O-O ----
225     # o processo todo e semelhante ao calculos dos BOOPs para todas
    particulas,
226     # independente do tipo de ligacao. A diferenca e que sao considerados
    somente as
227     # posicoes e a relacao de vizinhanca entre particulas de alcool (O)
228     # ---------
```

```
229     # caso o sistema seja composto por somente agua, o vetor de zeros para
    definir os
230     # BOOPs das ligacoes O-O
231     if(len(positions_O) == 0):
232       qs_O.append(aux_0)
233       qs_avg_O.append(aux_0)
234       qs_avg_avg_O.append(aux_0)
235       if(int(l) % 2 == 0):
236         ws_O.append(aux_0)
237         ws_avg_O.append(aux_0)
238         ws_avg_avg_O.append(aux_0)
239     # caso contratio, o calculo e realizado normalmente
240     else:
241       ngbs = vor.compute((box, positions_O)).nlist
242       q = fd.order.Steinhardt(l = l)
243       q_part = q.compute((box,positions_O), ngbs).particle_order
244       qs_O.append(q_part)
245       qavg = fd.order.Steinhardt(l = l, average = True)
246       qavg_part = qavg.compute((box,positions_O), ngbs).particle_order
247       qs_avg_O.append(qavg_part)
248
249       ngbs_indices_list = [[i] for i in range(len(positions_O))]
250       for i, j in ngbs[:]:
251         if(i == ngbs_indices_list[i][0]):
252           ngbs_indices_list[i].append(j)
253
254       avg_aux = [[] for i in range(len(positions_O))]
255       qavg_avg_part = [0 for i in range(len(positions_O))]
256
257       for i in range(len(ngbs_indices_list)):
258         for j in range(len(ngbs_indices_list[i])):
259           avg_aux[i].append(qavg_part[ngbs_indices_list[i][j]])
260         qavg_avg_part[i] = mean(avg_aux[i])
261
262       qs_avg_avg_O.append(qavg_avg_part)
263
264       if(int(l) % 2 == 0):
265         w = fd.order.Steinhardt(l = l, wl = True)
266         w_part = w.compute((box,positions_O), ngbs).particle_order
267         ws_O.append(w_part)
268         wavg = fd.order.Steinhardt(l = l, wl = True, average = True)
269         wavg_part = wavg.compute((box,positions_O), ngbs).particle_order
270         ws_avg_O.append(wavg_part)
271
272         avg_aux = [[] for i in range(len(positions_O))]
273         wavg_avg_part = [0 for i in range(len(positions_O))]
274
```

```
275         for i in range(len(ngbs_indices_list)):
276           for j in range(len(ngbs_indices_list[i])):
277             avg_aux[i].append(wavg_part[ngbs_indices_list[i][j]])
278           wavg_avg_part[i] = mean(avg_aux[i])
279 
280         ws_avg_avg_O.append(wavg_avg_part)
281 
282 
283 # os parametros todos calculados (tanto qs quanto ws, suas medias e
     medias
284 # das medias, para todas as particulas independente da ligacao,
     considerando somente ligacoes
285 # agua-agua e considerando somente ligacoes alcool-alcool) sao
     armazenadas em uma mesma linha
286 # para cada particula do sistema, totalizando 135 colunas
287 
288 # as primeiras 45 colunas sao os qs, os ws, suas medias e as medias das
     medias
289 # para todas as particulas, independente do tipo de ligacao
290 for counter in range(len(positions)):
291   for l in range(10):
292     file_out_qs.write(str('{:14.12f}'.format(qs[l][counter])) + '   ')
293   for l in range(5):
294     file_out_qs.write(str('{:14.12f}'.format(ws[l][counter])) + '   ')
295   for l in range(10):
296     file_out_qs.write(str('{:14.12f}'.format(qs_avg[l][counter])) + '   '
    )
297   for l in range(5):
298     file_out_qs.write(str('{:14.12f}'.format(ws_avg[l][counter])) + '   '
    )
299   for l in range(10):
300     file_out_qs.write(str('{:14.12f}'.format(qs_avg_avg[l][counter])) + '
      ')
301   for l in range(5):
302     file_out_qs.write(str('{:14.12f}'.format(ws_avg_avg[l][counter])) + '
      ')
303 
304 
305   if(counter < len(positions_N)): # se a particula considerada e de agua
306     # as proximas 45 colunas sao os BOOPs considerando somente ligacoes
    do tipo N-N
307     for l in range(10):
308       file_out_qs.write(str('{:14.12f}'.format(qs_N[l][counter])) + '   '
    )
309     for l in range(5):
310       file_out_qs.write(str('{:14.12f}'.format(ws_N[l][counter])) + '   '
    )
```

```
311       for l in range(10):
312         file_out_qs.write(str('{:14.12f}'.format(qs_avg_N[l][counter])) + '
          ')
313       for l in range(5):
314         file_out_qs.write(str('{:14.12f}'.format(ws_avg_N[l][counter])) + '
          ')
315       for l in range(10):
316         file_out_qs.write(str('{:14.12f}'.format(qs_avg_avg_N[l][counter]))
       + '   ')
317       for l in range(5):
318         file_out_qs.write(str('{:14.12f}'.format(ws_avg_avg_N[l][counter]))
       + '   ')
319
320       # e as ultimas 45, referentes aos BOOPs de ligacoes O-O, sao
      preenchidas com zeros
321       for l in range(10):
322         file_out_qs.write(str(0) + '   ')
323       for l in range(5):
324         file_out_qs.write(str(0) + '   ')
325       for l in range(10):
326         file_out_qs.write(str(0) + '   ')
327       for l in range(5):
328         file_out_qs.write(str(0) + '   ')
329       for l in range(10):
330         file_out_qs.write(str(0) + '   ')
331       for l in range(5):
332         file_out_qs.write(str(0) + '   ')
333
334     else: # ja se uma particula de alcool e considerada
335       # as proximas 45 colunas, referentes aos BOOPs considerando somente
      ligacoes do tipo N-N,
336       # sao preenchidas com zeros
337       for l in range(10):
338         file_out_qs.write(str(0) + '   ')
339       for l in range(5):
340         file_out_qs.write(str(0) + '   ')
341       for l in range(10):
342         file_out_qs.write(str(0) + '   ')
343       for l in range(5):
344         file_out_qs.write(str(0) + '   ')
345       for l in range(10):
346         file_out_qs.write(str(0) + '   ')
347       for l in range(5):
348         file_out_qs.write(str(0) + '   ')
349
350       # e as ultimas 45 colunas sao os BOOPs considerando somente ligacoes
      do tipo O-O
```

```
351       for l in range(10):
352         file_out_qs.write(str('{:14.12f}'.format(qs_O[l][counter-len(
    positions_N)])) + '   ')
353       for l in range(5):
354         file_out_qs.write(str('{:14.12f}'.format(ws_O[l][counter-len(
    positions_N)])) + '   ')
355       for l in range(10):
356         file_out_qs.write(str('{:14.12f}'.format(qs_avg_O[l][counter-len(
    positions_N)])) + '   ')
357       for l in range(5):
358         file_out_qs.write(str('{:14.12f}'.format(ws_avg_O[l][counter-len(
    positions_N)])) + '   ')
359       for l in range(10):
360         file_out_qs.write(str('{:14.12f}'.format(qs_avg_avg_O[l][counter-
    len(positions_N)])) + '   ')
361       for l in range(5):
362         file_out_qs.write(str('{:14.12f}'.format(ws_avg_avg_O[l][counter-
    len(positions_N)])) + '   ')
363
364     # entao passa-se para a proxima linha (particula) e as novas 135
    colunas sao preenchidas
365     # seguindo a mesma logica
366     file_out_qs.write('\n')
```

Já o algoritmo em python onde a rede neural é definida e treinada é exposto abaixo.

```
1  # pacotes utilizados no codigo sao importados
2  import numpy as np
3  import pandas as pd
4  from sklearn.model_selection import train_test_split
5  import tensorflow as tf
6
7  # o dataset, ja organizado com os arquivos de BOOPs referentes as 30
       configuracoes
8  # usadas para treino/validacao, e armazenado
9  dataset = pd.read_csv('dataset3.dat', sep = '\s+', header = None, usecols =
       columns)
10 X = dataset.iloc[:,:].values # os parametros sao amarzenados no vetor X
11 y = np.array((), dtype = int) # o vetor y vazio e criado
12 for j in range(3): # para as tres misturas diferentes
13   # o arquivo de entrada e definido de forma que
14   # as primeiras 2mil particulas do arquivo sao de configuracoes na fase 0
15   for i in range(2000):
16     y = np.append(y, [0])
17   # as proximas 2mil na fase 1
18   for i in range(2000):
19     y = np.append(y, [1])
20   # etc
```

```
  for i in range(2000):
    y = np.append(y, [2])
  for i in range(2000):
    y = np.append(y, [3])
  for i in range(2000):
    y = np.append(y, [4])

# especificamente, temos que as fases sao
# 0 -> I (BCC)
# 1 -> II (HCP)
# 2 -> III (amorfo)
# 3 -> LDL
# 4 -> HDL

# o conjunto de dados e dividido nos vetores de atributos (X) de treino e validacao (_train e _test)
# e nos vetores de rotulos (y) de treino e validao, de forma que a validacao e 10% do total de pontos
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1, random_state = 3)

# usando tensorflow.keras, o objeto classifier e criado a partir da classe Sequential
classifier = tf.keras.models.Sequential()
# a camada de entrada, com 135 nos, e ligada a primeira camada escondida, com 180 nos
# essa ultima com funcao de ativacao ReLu, inicializada usando glorot e com bias=True
classifier.add(tf.keras.layers.Dense(units=180,use_bias=True, kernel_initializer='glorot_uniform',activation='relu',input_dim = 135))
# a segunda camada escondida, com 90 nos e adicionada com os mesmos hiperparametros
classifier.add(tf.keras.layers.Dense(units=90,use_bias=True, kernel_initializer='glorot_uniform',activation='relu'))
# outra camada com 30 nos e adicionada
classifier.add(tf.keras.layers.Dense(units=30,use_bias=True, kernel_initializer='glorot_uniform',activation='relu'))
# a camada de saida, com 5 nos e adicionada, com funcao de ativacao softmax
classifier.add(tf.keras.layers.Dense(units = 5,use_bias=True, kernel_initializer='glorot_uniform',activation='softmax'))
# a rede neural e compilada e o processo de ajuste dos dados de saida aos de entrada (fit) e realizado para 40 epochs
classifier.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics = ['accuracy'])
classifier.fit(X_train, y_train, batch_size = 32, epochs = 40)

# os valores do conjunto de validacao sao preditos e definise como
```

```
    resultado o maior valor da camada de saida
y_pred = classifier.predict(X_test)
y_pred2 = np.zeros((3000,), dtype = int)
for i in range(len(y_pred)):
  yaux = np.argsort(y_pred[i])
  y_pred2[i] = int(yaux[4])
# acuracia de validacao e calculada
a = accuracy_score(y_pred2,y_test)
print('Accuracy is:', a*100)
```

Com a rede neural treinada, pode-se utilizar o objeto *classifier* para predizer (usando o método *predict*) a fase de todos os vetores $\mathbf{q}$, referentes à cada partícula de cada configuração das diferentes misturas e água pura.

5.2 Apêndice B: produção resultante da dissertação

Phase classification using neural networks: application to supercooled, polymorphic core-softened mixtures

Vinicius F. Hernandes[a,1], Murilo S. Marques[b], José Rafael Bordin[c,2]

[a] *Programa de Pós-Graduação em Física, Departamento de Física, Instituto de Física e Matemática, Universidade Federal de Pelotas. Caixa Postal 354, 96001-970, Pelotas-RS, Brazil.*

[b] *Centro das Ciências Exatas e das Tecnologias, Universidade Federal do Oeste da Bahia Rua Bertioga, 892, Morada Nobre, CEP 47810-059, Barreiras-BA, Brazil*

[c] *Departamento de Física, Instituto de Física e Matemática, Universidade Federal de Pelotas. Caixa Postal 354, 96001-970, Pelotas-RS, Brazil.*

Abstract

Characterization of phases of soft matter systems is a challenge faced in many physicochemical problems. For polymorphic fluids it is an even greater challenge. Specifically, glass forming fluids, as water, can have, besides solid polymorphism, more than one liquid and glassy phases, and even a liquid-liquid critical point. In this sense, we apply a neural network (NN) algorithm to analyze the phase behavior of a core-softened mixture of core-softened CSW fluids that have liquid polymorphism and liquid-liquid critical points, similar to water. We also apply the NN to mixtures of CSW fluids and core-softened alcohols models. We combine and expand two methods based on bond-orientational order parameters to study mixtures, applied to mixtures of hardcore fluids by Boattini and co-authors [*Molecular Physics 116, 3066-3075 (2018)*] and to supercooled water by Martelli and co-authors [*The Journal of Chemical Physics 153, 104503 (2020)*], to include longer range coordination shells. With this, the trained neural network (NN) was able to properly predict the crystalline solid phases, the fluid phases and the amorphous phase for the pure CSW and CSW-alcohols mixtures with high efficiency. More than this, information about the phase populations, obtained from the NN approach, can help verify if the phase transition is continuous or discontinuous, and also to interpret how the metastable amorphous region spreads along the stable high density fluid phase. These findings help to understand the behavior of supercooled polymorphic fluids and extend the comprehension of how amphiphilic solutes affect the phases behavior.

Keywords:

[1] vfhernandes@ufpel.edu.br
[2] jrbordin@ufpel.edu.br

1. Introduction

In the last decade, machine learning (ML) models successfully penetrated into virtually all areas of the scientific community, no longer being considered only an object of study *per se*, but also a tool that can help solve the more diverse kind of problems faced by scientists [1, 2, 3, 4, 5]. Some few examples of ML applications in physics, chemistry and materials science include molecular and atomistic simulation [6, 7], self-assembly of molecules [8, 9, 10, 11], force fields parametrization [12, 13, 14], soft-materials and proteins engineering [15, 16] and drug discovery [17]. Another application that has recently being perfected and improved, taking advantage of ML models, is phase recognition. The task of identifying the structural formation of matter from local arrangements obtained from simulation data can be significantly refined with the utilization of statistical learning techniques. This approach is showing excellent results, such as local structure detection of polymorphic systems with supervised [18] and unsupervised [19] learning, identification of soft matter [20] and amorphous materials [21] structures using convolutional neural networks, and phase prediction of high-entropy alloys [22, 23].

A system that is constantly under investigation, given its complexity, and that has been greatly benefited from a statistical approach, is water and its mixtures. Water is the solvent of life, and the main solvent in industry. Also, pure and "simple" water presents more than 70 known anomalies [24], making it unique [25]. The origin of the high number of anomalies for temperatures in the supercooled regime can be related to a two liquids coexistence line that ends in a liquid-liquid critical point (LLCP), and to the competition between these liquids [26, 27, 28]. At low densities, in the low density liquid (LDL) phase, the water molecules have an ordered tetrahedral structure, while the high density liquid (HDL) state is characterized by a more distorted tetrahedral structure, and with local higher density as consequence. The fact that water itself is a mixture of two liquids was hypothesized in the 90s, with an extensive theoretical debate since then, specially in the last decade [29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40]. Currently, the main theoretical evidences indicate that the LLCP exists, and the experimental evidences that support this conclusion are growing in the last years [41, 42, 43, 44, 45, 46, 47]. However, is an extremely hard task to achieve experimentally this region, known as "no man's land", due to rapidly crystallization. In this sense, a computational approach is particularly useful. An extensive number of works have studied the region near LLCP with molecular simulations using different potentials and approaches [34, 40, 26, 48]. More recently, ML algorithms are being utilized for recognizing the structures exhibited by water, specially nearby the second critical point. Distinct supervised learning approaches with Neural Networks are being explored, some of them based on bond-orientational order parameters [49] or symmetry functions [18], others using data obtained from ab-initio calculations [50, 51], or even networks with more complex architectures, combining different methods [52]. Other works studied the relation between structure and dynamics in the same supercooled region, but for general liquids and glasses, using unsupervised methods [53, 54].

Along with research regarding pure water systems, a set of works have focused on the study of aqueous solutions in the supercooled regime. For instance, in the experimental work by Zhao and Angell [55], the crystallization was repressed in the no man's land by adding ionic liquids that dissolve ideally in water, preventing the crystallization without destroying the water anomalies. Another class of solutions, which are simpler if compared to more complex systems, is the one of short-chain alcohols in water. It can be treated as a binary system, facilitating the computational approach. The motivation behind the studies of these particular systems lays on their wide range of application such as dispersion media [56], disinfectant [57], in the food [58] and medical [59] industries, among others. Many experimental and theoretical works have studied these short-chain alcohol/water mixtures [60, 61, 62, 63, 64, 65, 66]. In our recent works, we have performed Molecular Dynamics (MD) simulations with a core-softened (CS) approach to investigate the behavior of methanol-water [67] and water mixtures with methanol, ethanol and 1-propanol [68]. In this approach, the waterlike solvent is modeled as the CSW fluid proposed by Franzese [69]. Although it is a spherically core-softened potential with two length scales, without any directionallity and, therefore, is not water [70], this CS approach has been largely employed to understand water anomalies both in bulk and confined environments [71, 72, 73, 74]. Particularly, the CSW model that we use in our work is able to reproduce the anomalous behavior of water in the supercooled regime, including the existence of two liquid phases whose coexistence line ends in the LLCP. Based on this potential, the alcohols are modeled as rigid polymers, as proposed by Urbic and co-authors [75]. Moreover, the hydroxyl group is modeled as a CSW bead, while the hydrophobicity of the polar sites is given by a modified Lennard Jones (LJ) potential. It was found [68] the addition of distinct concentrations of alcohols with distinct chain lengths lead to the suppression of the crystal phase, with the favoring of the amorphous phase and the existence of the liquid-liquid phase transition - in addition to the waterlike anomalies in the supercooled regime. With a particular interest for this present work, we observe a variety of phases: a high-density liquid (HDL) and a low-density liquid phase (LDL), two solid phases: a body-centered cubic one (BCC - phase I) and a hexagonal close-packed phase (HCP - phase II), and an amorphous solid phase (phase III). This polymorphism makes these mixtures great candidates to test if a NN algorithm is able to recognize the distinct phases in water-solvent mixtures in the supercooled regime.

Additionally, contrasting with the case of pure-water [52, 49] there is a lack of works applying ML models to autonomously recognize the phases of water-alcohol mixtures near the LLCP, an approach that can help to better understand the structural behavior of these systems. Given that, in this work, we set up a Neural Network based on Steinhardt parameters, [76, 77] adapted for binary mixtures, similar to what has been done by Boattini *et al.* [78], capable of identify the phases of (methanol/ethanol/1-propanol)-water mixtures for different alcohol concentrations. Our goal is to check if this NN based approach can properly predict the phases and phase transitions and provide new insights about the polymorphism in the supercooled regime.

The paper is structured as follows. In Section, 2 we describe the NN architecture and the parameters utilized for map the molecules' local structures. The results obtained, namely the phase diagram predicted by the NN for the different mixtures and concentrations, alongside a population analysis (the number of particles in each phase for a specific pressure-temperature configuration) are presented in Section 3. In Section 4, we present a closing discussion, with the principal remarks and some perspectives for new works.

2. The Computational Details

The NN approach requires only a system snapshot. The last configuration from each (N, P, T) simulation performed in our last work [68] was chosen as input for the NN. All the systems, for different alcohols and concentrations and pure water, are composed by $N = 1000$ molecules. Water and alcohol's hydroxyl groups are modeled as CSW particles, while the hydrophobic carbon chain in alcohols molecules are LJ sites. The waterlike solvent is monomeric, while the alcohols are linear rigid polymers: methanol is modeled as a dumbbell – one CSW site, one LJ site, ethanol as a trimer – one CSW site, two LJ sites – and propanol as a tetramer – one CSW site, three LJ sites. Detailed information about the simulation methods, parameters and the models can be found in our previous work [68]. For the pure solvent and all mixtures cases, we have observed five distinct phases. Two liquid phases, namely Low Density Liquid (LDL) and High Density Liquid (LDL), separated by a first order coexistence line that ends in the LLCP. Also, above the LLCP, the Widom Line (WL) delimits the border between the LDL-predominant and HDL-predominant regions. Two crystaline phases were observed. At lower pressures, the system is in a BCC phase, that will be called of solid I, or just I for simplification, and in a HCP – or solid phase II – at intermediate pressures. Finally, at higher pressures, the system has an amorphous solid phase, or phase III. We called it an amorphous solid once it is disordered, with a structure similar to the HDL phase, but with no diffusion. The amorphous-HDL transition was characterized by an increase in the diffusion constant and by maxima in the isobaric expansion coefficient and in the specific heat at constant pressure [68]. All quantities with an asterisk are in reduced dimensionless units [79].

To map the local environment of each particle of the system into a vector used as input for the ML model, a series of Bond-Order Parameters (BOOP) [76, 77] are calculated. The last configuration from the (N, P, T) simulations are the input for the freud analysis python package [80] to calculate the BOOPs of the 1000 CSW particles in the system. The Voronoi tessellation was used to define nearest neighbors. For each particle i with $N_b(i)$ neighbors, first we

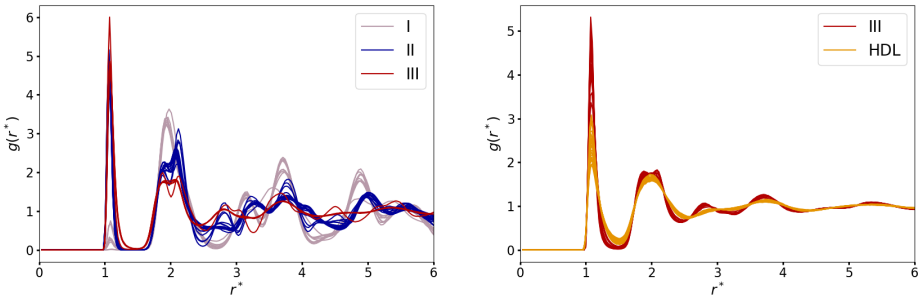
calculate $q_{lm}(i)$, $\bar{q}_{lm}(i)$ and $\bar{\bar{q}}_{lm}(i)$, defined as

$$\begin{aligned} q_{lm}(i) &= \frac{1}{N_b(i)} \sum_{j=1}^{N_b(i)} Y_{lm}(\vec{r}_{ij}), \\ \bar{q}_{lm}(i) &= \frac{1}{N_b(i)+1} \sum_{k \in \{i, N_b(i)\}} q_{lm}(k), \\ \bar{\bar{q}}_{lm}(i) &= \frac{1}{N_b(i)+1} \sum_{k \in \{i, N_b(i)\}} \bar{q}_{lm}(k), \end{aligned} \tag{1}$$

where $Y_{lm}(\vec{r}_{ij})$ are the spherical harmonics for the distance vector $\vec{r}_{ij}$ separating particle i from j. Then, from Equation (1) we have $q_l(i)$, $\bar{q}_l(i)$ and $\bar{\bar{q}}_l(i)$, given by

$$\begin{aligned} q_l(i) &= \sqrt{\frac{4\pi}{2l+1} \sum_{m=-l}^{l} |q_{lm}(i)|^2}, \\ \bar{q}_l(i) &= \sqrt{\frac{4\pi}{2l+1} \sum_{m=-l}^{l} |\bar{q}_{lm}(i)|^2}, \\ \bar{\bar{q}}_l(i) &= \sqrt{\frac{4\pi}{2l+1} \sum_{m=-l}^{l} |\bar{\bar{q}}_{lm}(i)|^2}, \end{aligned} \tag{2}$$

where $q_l(i)$ is the rotational invariant BOOP, $\bar{q}_l(i)$ its average and $\bar{\bar{q}}_l(i)$ the average of the averages. The average values make it possible to get information about, approximately, the second-shell neighbors. Here, we introduced the average-average parameter, $\bar{\bar{q}}_l(i)$, which holds information regarding the structure of, approximately, third-shell neighbors. We want to test if this further parameter can be useful to uniquely identify a local structure that, for different configurations within the same phase, shows significant differences in the long-range coordination shells, as can be seen by analyzing the radial distribution function $g(r^*)$ for pure water with $T^* = 0.26$ at distinct pressures, in Fig. 1(a) and at distinct temperatures with fixed pressure equals to $P^* = 0.28$, in Fig. 1(b). The former case is that of a fixed temperature at which the three solid phases occur. The latter is the case of a fixed pressure at which the system presents an amorphous-HDL transition.
Then, the cubic BOOPs, $w_l(i)$, their average, $\bar{w}_l(i)$, and the average of the



(a) (b)

Figure 1: (a) Radial distribution function for pure water with $T^* = 0.26$. Grey lines are the pressures in the BCC phase, blue lines in the HCP phase and red in the amorphous phase. (b) Radial distribution function for pure water with $P^* = 0.28$. Red lines are the temperatures in the amorphous phase and yellow lines in the HDL phase.

average, $\bar{w}_l(i)$,

$$w_l(i) = \frac{\sum_{m_1+m_2+m_3=0} \begin{pmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{pmatrix} q_{lm_1}(i) q_{lm_2}(i) q_{lm_3}(i)}{\left(\sum_{m=-l}^{l} |q_{lm}(i)|^2\right)^{3/2}},$$

$$\bar{w}_l(i) = \frac{\sum_{m_1+m_2+m_3=0} \begin{pmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{pmatrix} \bar{q}_{lm_1}(i) \bar{q}_{lm_2}(i) \bar{q}_{lm_3}(i)}{\left(\sum_{m=-l}^{l} |\bar{q}_{lm}(i)|^2\right)^{3/2}}, \tag{3}$$

$$\bar{\bar{w}}_l(i) = \frac{\sum_{m_1+m_2+m_3=0} \begin{pmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{pmatrix} \bar{\bar{q}}_{lm_1}(i) \bar{\bar{q}}_{lm_2}(i) \bar{\bar{q}}_{lm_3}(i)}{\left(\sum_{m=-l}^{l} |\bar{\bar{q}}_{lm}(i)|^2\right)^{3/2}},$$

are calculated. Here, the term in parentheses corresponds to the Wigner $3j$ symbol. The set of parameters described so far should give a vector, composed by translationally and rotationally invariants, capable to uniquely describe the different phases' local environment. However, since we are dealing with binary-mixtures systems, a distinction between the parameters related to water molecules and those related to alcohol molecules. is needed, specifically for the hydroxyl group of the alcohols' molecules, since they are modeled by the same potential interaction and parameters. This common factor makes this approach for binary mixtures expandable to other alcohols, as long as the hydroxyl group is explicit in the simulation, since the bonds are always calculated considering the water-hydroxyl interaction. Given that, we consider three variations for each parameter calculated. For instance, instead of a unique $q_l(i)$, we have

$q_l^{W-A/A-W}(i)$, $q_l^{W-W}(i)$ and $q_l^{A-A}(i)$. The first term is the $q_l(i)$ parameter for i being a water (alcohol) molecule considering all type of neighbors, water (alcohol) or alcohol (water). The second term is the $q_l(i)$ where i is a water molecule, and only water molecules are considered as neighbors when performing the calculation. The last one is the parameter for an alcohol molecule, only considering alcohols molecules as neighbors.

Finally, we have a vector $\mathbf{q}(i)$ for each particle i, composed by all the different BOOPs and cubic BOOPs (and their averages), taking into consideration the variations applied for the binary-mixture case. For any molecule we have

$$\begin{aligned}\mathbf{q}(i) = \Big(&\big\{q_l^{W-A}(i)\big\}, \big\{\bar{q}_l^{W-A}(i)\big\}, \big\{\bar{\bar{q}}_l^{W-A}(i)\big\}, \big\{q_l^{W-W}(i)\big\}, \big\{\bar{q}_l^{W-W}(i)\big\}, \big\{\bar{\bar{q}}_l^{W-W}(i)\big\},\\ &\big\{q_l^{A-A}(i)\big\}, \big\{\bar{q}_l^{A-A}(i)\big\}, \big\{\bar{\bar{q}}_l^{A-A}(i)\big\}, \big\{w_l^{W-A}(i)\big\}, \big\{\bar{w}_{l'}^{W-A}(i)\big\}, \big\{\bar{\bar{w}}_{l'}^{W-A}(i)\big\},\\ &\big\{w_{l'}^{W-W}(i)\big\}, \big\{\bar{w}_{l'}^{W-W}(i)\big\}, \big\{\bar{\bar{w}}_{l'}^{W-W}(i)\big\}, \big\{w_{l'}^{A-A}(i)\big\}, \big\{\bar{w}_{l'}^{A-A}(i)\big\}, \big\{\bar{\bar{w}}_{l'}^{A-A}(i)\big\}\Big)\end{aligned}$$

with $l \in [3, 12]$ and l' assuming only the even values of l. For a water molecule, all the parameters with A-A in the exponent are equal to zero. If we consider an alcohol molecule, we change W-A for A-W in Equation 4 and set all parameters with W-W in the exponent to zero.

Thus, $\mathbf{q}(i)$ is a 135-dimensional vector that uniquely identify the local structure of the systems' particles up to the third-shell neighbors, distinguishing water molecules from alcohol molecules.

The model chosen to autonomously identify the local environments is a Feed-Forward Neural Network, with the vector in Eq. 4 as the Input Layer (IL), three Hidden Layers (HL) with 180, 90 and 30 neurons, respectively, and a 5-dimensional Output Layer (OL), where each one of the output neurons represents the probability of a particle being in one of the five possible phases the systems analyzed can assume. The whole NN approach was performed using the keras [81] package, with TensorFlow [82] backend. Glorot initialization [83] was applied for all layers; Rectifier Linear Unit was used as activation function for the IL and the HLs, whereas Softmax was used as activation function for the OL; sparse categorical crossentropy was used as loss function with adam [84] as optimizer. The training was performed for 40 epochs, using a batch size equals to 32. Different network's architecture were tested, as well as hyperparameters, and the setting described presented the best validation accuracy overall.

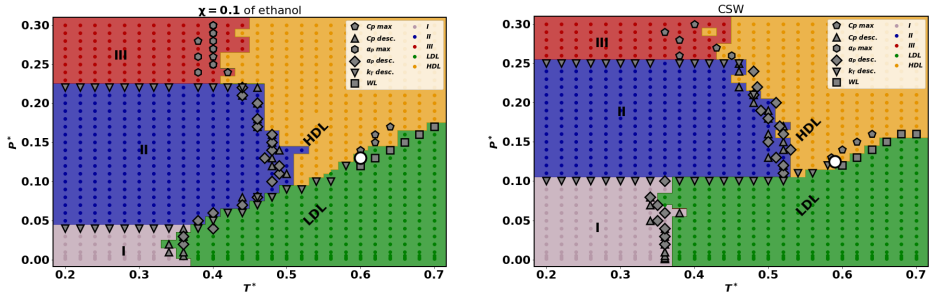
To efficiently teach the network to distinguish the different phases, we used as input a $\mathbf{q}(i)$ vector for each CSW particle of a system, from a total of thirty configurations, two for each phase and for each mixture with alcohol concentration equals to 10%. Since the number of particles per system is equal to 1000, the total number of data points for the train set is 30 thousand, each one labeled with one of the five possible phases. Since the snapshots with the particle's positions are taken from systems which are in equilibrium, we assume it is uniform, with all the N particles in the same phase. So, the phase used for label train data is already known, as it was found from the thermodynamic analysis

performed in our previous work [68]. For example, if we use the N particles from the ethanol-water mixture with concentration $\chi = 0.10$, for temperature $T = 0.20$ and $P = 0.01$, as part of the train set, we use our previous analysis to label the N particles with 'phase I'.

Afterwards, we use the trained NN to predict the phase of the particles from all the configurations analyzed in [68], i.e. the three different mixtures with three different concentrations, besides pure water. The results regarding the predicted phase for each particle give two different information. The first one is the system's phase, which is simply found by analyzing the dominant phase in a single system. The second one is the type of phase transition occurring near the transition curves, which can be extracted from a population analysis (the number of particle in each phase for a single system) . With the thermodynamic analysis, transitions points occur where response functions present a maximum or a discontinuity. First and second order phase transitions can be distinguished analysing the response functions, which result in the transitions points represented in Fig. 2 as grey points. Moreover, the population behavior can give more insights about the system phase behavior in the supercooled regime.

3. Results and discussion

The neural network achieved accuracy up to 99% and 99.3%, for training (see Supplementary Material - SM) and validation, respectively. Similar results can also be achieved with one less hidden layer, but increasing the epochs. Overall, it was the faster method to maintain a higher number of hidden layers, but train with less epochs.



(a) (b)

Figure 2: Phase diagrams of (a) water-ethanol mixture with concentration $\chi = 0.1$ and (b) pure water, predicted by the neural network. The grey region points were classified as solid phase I, blue region dots as solid phase II, red region dots as amorphous phase III, green region dots as LDL and the yellow region dots as in the HDL phase. The grey markers with distinct shapes stands for discontinuities and maxima in the response functions isothermal compressibility κ_T, isobaric expansion coefficient α_P and specific heat at constant pressure C_P. The white circle is the LLCP obtained in the Ref. [68].

In Fig. 2(a) is presented the phase diagram obtained with the neural network for the water-ethanol mixture with concentration of ethanol equals to 10%.

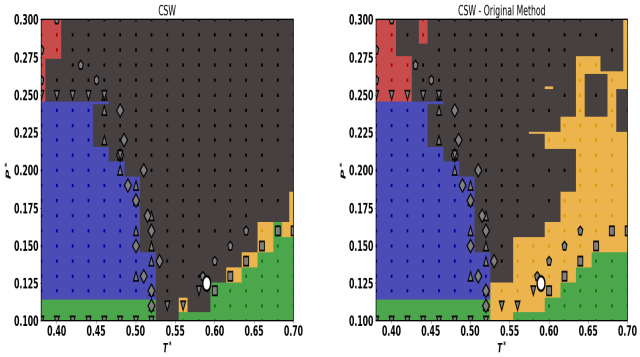
This is one of the systems from which two snapshots per phase were used to train the NN. Here, each colored circle corresponds to one configuration with defined pressure, temperature and phase, being each phase represented by a unique color. The grey markers correspond to the transition points found from the thermodynamic analysis, forming the "ground-truth" transition lines that delimit each phase and the white circle is the LLCP. We notice from the figure that there is a great overall correspondence between the phases predicted by the network and by the classical approach. That behavior is also noticed for pure CSW, Fig. 2(b), and all the combinations of mixture and concentration, shown in the SM. The overall accuracy, defined as the number of configurations with defined temperature and pressure that the network correctly predicted the phase divided by the total number of configurations, are shown in Table 1. Here we note that the overall accuracy found when predicting the phase diagrams is significantly lower than the train/validation accuracy. That is explained by the configurations near the transitions lines which phases were incorrectly classified by the network. Also, as it can be seen in the Supplementary Material, the accuracy is already high in the first epochs of the training process, and then increases gradually, indicating that the network learns the main model's features quickly. Also, is important to address that the randomness in choosing the systems for the train dataset plays a major role in the accuracy. For instance, if a random point near the phases boundaries is chosen, the agreement is worst once in these regions we can have a mixture of two phases, what leads to a bad train dataset.

Table 1: Accuracy for pure water and all combinations of mixtures.

	pure water	methanol			ethanol			propanol		
χ	-	0.01	0.05	0.1	0.01	0.05	0.1	0.01	0.05	0.1
Accuracy	0.94	0.93	0.92	0.94	0.89	0.92	0.93	0.90	0.94	0.92

The compatibility between the methods is particularly interesting in the supercritical region, for temperatures higher than the critical temperature. In this region, we cannot precisely define two different liquid phases, but one liquid phase with two different characteristics, one closely related to the HDL phase and the other to the LDL phase. The separation within supercritical liquid phase is given by the Widom Line (WL), an extension of the LLPT curve into the one-phase region and the locus of maximum fluctuations of the order parameter [85, 86], represented by the grey squares in the diagram. The NN successfully separates the HDL-like supercritical liquid from the LDL-like one, and the HDL and LDL liquids in the subcritical region. To check if the system is crossing the phase coexistence line or the WL, we can analyze the isothermal populations. This population analysis is also useful to analyze the amorphous-HDL boundary region, where the larger discrepancy between the neural network approach and the thermodynamic analysis was observed. At this point, is important to recall that the amorphous phase is not an equilibrium one, and the amorphous-HDL boundary may change if the system is going through a cooling

or a heating process – our results were evaluated by a cooling process. The amorphous phase has a smaller diffusion constant – $D \approx 0$ – compared to the HDL phase, what indicates absence of movement - for this reason, we are calling it a "solid". Also, the maxima in α_P and C_P, shown in the phase diagrams (Fig. 2) and the smooth change in the structural parameters, as the pair excess entropy s_2, suggest a boundary between the amorphous and HDL phases [68]. Once the HDL and the amorphous phase have similar short-range ordering [68], we expand the method by Martelli and co-authors [49] to include structural information about the third-shell neighbors, still considering different parameters for different molecule, as done by Boattini et al. [78] . It is important for our case once the waterlike characteristics of core-softened fluids can be related to competitions in the long range coordination shells - not only in the first or second one [87, 88, 68]. To this end, we include the average-average therms of Eqs. 2 and 3. This approach leads to a slightly better agreement between the NN method and the analysis based in the response functions. However, the results are significantly distinct when we look at the population of particles in a particular phase for each point in the phase diagram.

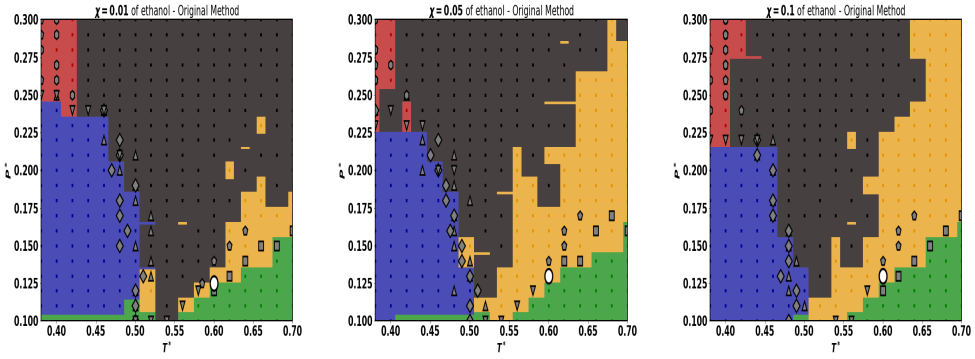


(a) (b)

Figure 3: Phase diagrams predicted using BOOPs and their average (a) and including the average-average parameters (b) for the pure CSW fluid. Blue dots were classified as phase II, red as phase III, green as LDL, yellow as HDL and the black dots indicates the existence of amorphous-like particles in the HDL phase.

Each point in the phase diagram was defined using the population analysis: if 50% or more of the particles are in one of the five possible phases, per say HDL phase, the point is classified as HDL. Nevertheless, some particle in the HDL phase can have populations classified as a distinct phase. As an example, we evaluate the population of amorphous particles in points that were classified in the stable HDL phase, since it can indicate how the metastable region spreads in the phase diagram. In the Fig. 3 (a) and (b) we compare the case of pure CSW water using previously implemented methods [78, 49], adapted to our system, and our version, that includes longer correlations. The points in the metastable phase – which were classified as HDL but have at least one particle classified

as amourphous, are painted black, in contrast with the colors of the Fig. 2(b). While in the first phase diagram amorphous particles spread along practically every point in the HDL phase, in the second one the amorphous population occupy a smaller region. This indicates that including longer-range information will lead to a better classification of these glassy phases.



(a) (b) (c)

Figure 4: Phase diagrams predicted by the NN using average-average parameters, for $0.38 \leq \mathbf{T}^* \leq 0.70$ and $0.10 \leq \mathbf{P}^* \leq 0.30$ of water-ethanol mixture with (a) $\chi = 0.01$, (b) $\chi = 0.05$ and (c) $\chi = 0.1$. Blue dot regions were classified as phase II, red regions as phase III, green as LDL, yellow as HDL and the black region is the metastable phase (phase classified as LDL with particles classified as phase III).

We can also apply this analysis – with the average-average parameters – to see how the amorphous-like population changes as the concentration of alcohol in the solution increases. As we can see in the Fig. 4(a) for CSW-ethanol mixtures, and in the SM for the other alcohols, the region with amorphous population in the HDL phase increases for the lower concentration in comparison to the pure CSW fluid, Fig. 3(b), and then shrinks with the increase in concentration, as shown in Figs. 4(b) and (c). This agrees with our previous results [68], where we found that the low concentration of alcohol affects only the long range coordination shells. Then, as χ increases, it favors the short range organization, that becomes predominant and the long range effects are less relevant.

Using the NN classification, we can define how much particles are in one of the five phases. The number of particles defines the population of each phase. For instance, we can walk along an isotherm and see how the population changes. Taking the low temperature isotherm $T^* = 0.20$ for the ethanol mixture at $\chi = 0.1$, shown in the Fig. 5 (a), we can see that at the I-II transition practically all particles change from the BCC to the HCP structure. However, at the II-III transition, we can see fluctuations in the II and III populations from $P^* = 0.19$ to $P^* = 0.22$. Once this corresponds to a solid-amorphous transition, this is expected due the metastability of the phase III. After that, we can see the presence of HDL-like particles in the region III. Heating to $T^* = 0.50$, Fig. 5 (b), an isotherm that crosses liquid and solid phases, we can see the LDL-solid

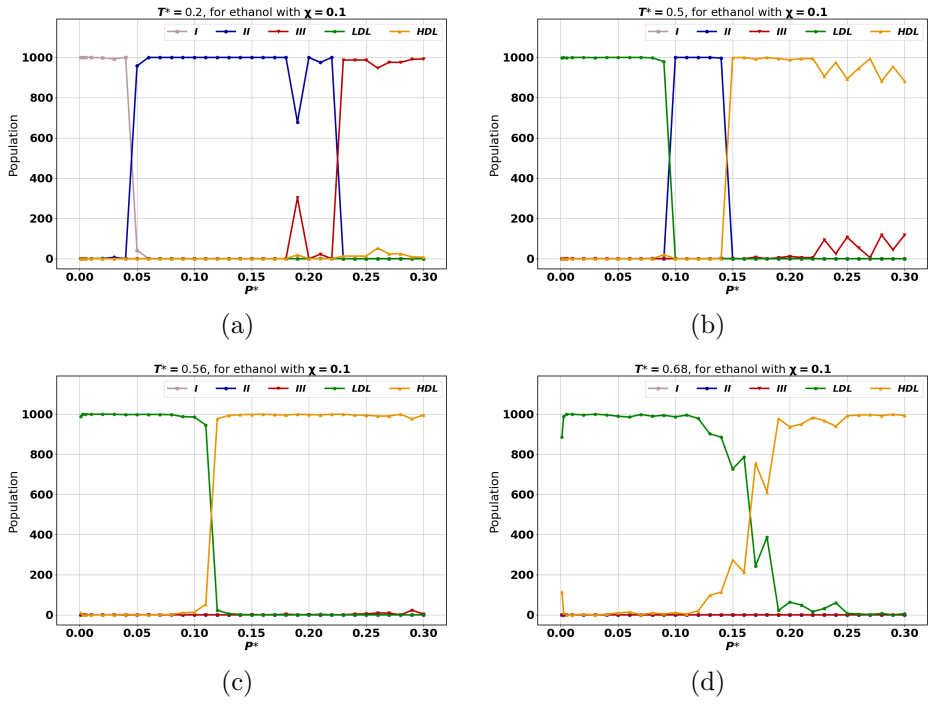


Figure 5: Population as a function of pressure for the water-ethanol mixture with concentration equals to 0.1 and for fixed temperature equals to (a) 0.2, (b) 0.5, (c) 0.56 and (d) 0.68.

II transition at $P^* = 0.10$ by the abrupt change in the populations of each one of the phases, same for the solid II-HDL transition at $P^* = 0.15$. Once again, the effect of the amorphous phase metastability has been noticed: at higher pressures the population of amorphous particles starts to increase. At the subcritical isotherm $T^* = 0.56$, shown in Fig. 5 (c), the LDL-HDL transition is clear at $P^* = 0.12$ – distinct from the supercritical isotherm $T^* = 0.68$, shown in Fig. 5 (d), where the transition from LDL-like to HDL-like behavior is continuous. Also, this temperature is high enough – and far enough from the metastable region – to ensure that there is no more amorphous-like particles in the system.

A similar analysis can be made along isobars. Here, we show the isobars $P^* = 0.01$, $P^* = 0.05$, $P^* = 0.14$ and $P^* = 0.29$ in Fig. 6(a) to (d), respectively. The first two show the abrupt change in solid and liquid populations for the solid I - LDL and solid II - LDL transitions. In the Fig. 6 (c) we can see the solid phase II to HDL transition at $T^* = 0.51$ followed by a change in the HDL-like and LDL-like populations as the isobar crosses the WL. Finnaly, we observe that the amorphous - HDL phase transition is smooth, with the particles structure gradually changing from one type to another until the high temperature limit,

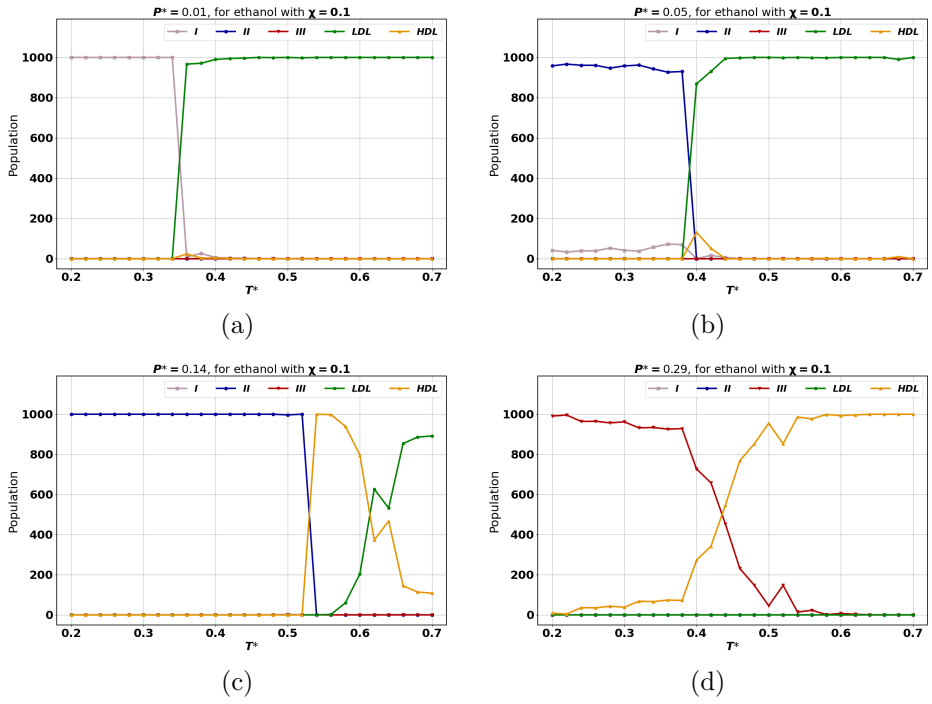


Figure 6: Population as a function of temperature for the water-ethanol mixture with concentration equals to 0.1 and for fixed pressures equals to (a) 0.01, (b) 0.05, (c) 0.14 and (d) 0.29.

where no more fluctuations from the metastable phases are observed.

4. Conclusion

In this paper, we have used a machine learning approach to classify phases of core softened CSW-alcohol mixtures, for different alcohols and concentrations, as well as pure CSW, in the supercooled regime. The neural network model inspired by the Refs. [78] and [49] uses an extensive set of unique bond-orientational order parameters for water-water, water-alcohol and alcohol-alcohol bonds, as input, and was extended to include longer-range coordination shells in comparison to the original method.

For pure CSW fluid and for all the possible combinations of mixtures and concentrations, the phase classification agrees with the thermodynamic analysis from the response functions [68]. The latter approach can be tiring and slow, needing an extensive calculation of physical variables to be analysed. Moreover, different variables have to be calculated and analysed for distinct transitions. Nevertheless, our method presents itself as a faster alternative, requiring always the same set of parameters to identify all the phases the systems can assume.

Additionally, since the model predicts phases of individual particles within a system, a population analysis can be performed, from which we showed it is possible to discern different kind of transitions (discontinuous or continuous transitions) and for the region where high and low density liquids appear, if a transition is taking place or the Widom Line is crossed.

The implementation applied is complementary to works that use a machine learning approach to study water in the supercooled regime, such as those in Refs. [52, 49], and explore a new applicability of the binary-mixture network developed in Ref. [78].

Data Availability

All data and codes used in this work are available upon reasonable request.

Authors Contribution

VFH worked on the the conceptualization, methodology, programming and software development, data acquisition and analysis, validation, writing of the original draft, revision and editing. MSM contributed with data acquisition and analysis, writing review and editing. JRB worked on the conceptualization, methodology, programming and software, data acquisition and analysis, writing review and editing, supervision, funding acquisition and project administration.

5. Acknowledgments

Without the public funding this research would not be impossible. VFH thanks the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Finance Code 001, for the MSc Scholarship. JRB acknowledge to Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Fundação de Apoio a Pesquisa do Rio Grande do Sul (FAPERGS) for financial support. All simulations were performed in the SATOLEP Cluster from the Group of Theory and Simulation in Complex Systems from UFPel.

References

[1] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. 91 (2019) 045002. doi:10.1103/RevModPhys.91.0450022.

[2] M. Buchanan, The power of machine learning, Nature Physics 15 (2019) 1208. doi:10.1038/s41567-019-0737-8.

[3] G. R. Schleder, A. C. M. Padilha, C. M. Acosta, M. Costa, A. Fazzio, From DFT to machine learning: recent approaches to materials science–a review, Journal of Physics: Materials 2 (3) (2019) 032001. doi:10.1088/2515-7639/ab084b.
URL https://doi.org/10.1088/2515-7639/ab084b

[4] G. B. Goh, N. O. Hodas, A. Vishnu, Deep learning for computational chemistry, Journal of Computational Chemistry 38 (16) (2017) 1291–1307. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.24764, doi:https://doi.org/10.1002/jcc.24764.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.24764

[5] A. C. Mater, M. L. Coote, Deep learning in chemistry, Journal of Chemical Information and Modeling 59 (6) (2019) 2545–2559. arXiv:https://doi.org/10.1021/acs.jcim.9b00266, doi:10.1021/acs.jcim.9b00266.
URL https://doi.org/10.1021/acs.jcim.9b00266

[6] F. Noé, A. Tkatchenko, K.-R. Müller, C. Clementi, Machine learning for molecular simulation, Annual Review of Physical Chemistry 71 (1) (2020) 361–390. doi:10.1146/annurev-physchem-042018-052331.

[7] M. Hellström, J. Behler, High-Dimensional Neural Network Potentials for Atomistic Simulations, Ch. 3, pp. 49–59. arXiv:https://pubs.acs.org/doi/pdf/10.1021/bk-2019-1326.ch003, doi:10.1021/bk-2019-1326.ch003.
URL https://pubs.acs.org/doi/abs/10.1021/bk-2019-1326.ch003

[8] A. W. Long, A. L. Ferguson, Nonlinear machine learning of patchy colloid self-assembly pathways and mechanisms, The Journal of Physical Chemistry B 118 (15) (2014) 4228–4244, pMID: 24660984. doi:10.1021/jp500350b.

[9] W. F. Reinhart, A. W. Long, M. P. Howard, A. L. Ferguson, A. Z. Panagiotopoulos, Machine learning for autonomous crystal structure identification, Soft Matter 13 (2017) 4733–4745. doi:10.1039/C7SM00957G.

[10] X. Zhao, C. Liao, Y.-T. Ma, J. B. Ferrell, S. T. Schneebeli, J. Li, Top-down multiscale approach to simulate peptide self-assembly from monomers, Journal of Chemical Theory and Computation 15 (3) (2019) 1514–1522, pMID: 30677300. arXiv:https://doi.org/10.1021/acs.jctc.8b01025, doi:10.1021/acs.jctc.8b01025.
URL https://doi.org/10.1021/acs.jctc.8b01025

[11] C. S. Adorf, T. C. Moore, Y. J. U. Melle, S. C. Glotzer, Analysis of self-assembly pathways with unsupervised machine learning algorithms, The Journal of Physical Chemistry B 124 (1) (2020) 69–78, pMID: 31813215. arXiv:https://doi.org/10.1021/acs.jpcb.9b09621, doi:10.1021/acs.jpcb.9b09621.
URL https://doi.org/10.1021/acs.jpcb.9b09621

[12] Y. Li, H. Li, F. C. Pickard, B. Narayanan, F. G. Sen, M. K. Y. Chan, S. K. R. S. Sankaranarayanan, B. R. Brooks, B. Roux, Machine learning force field parameters from ab initio data, Journal of Chemical Theory and Computation 13 (9) (2017) 4492–4503, pMID: 28800233. arXiv:https://

doi.org/10.1021/acs.jctc.7b00521, doi:10.1021/acs.jctc.7b00521.
URL https://doi.org/10.1021/acs.jctc.7b00521

[13] V. Botu, R. Batra, J. Chapman, R. Ramprasad, Machine learning force fields: Construction, validation, and outlook, The Journal of Physical Chemistry C 121 (1) (2017) 511–522. arXiv:https://doi.org/10.1021/acs.jpcc.6b10908, doi:10.1021/acs.jpcc.6b10908.
URL https://doi.org/10.1021/acs.jpcc.6b10908

[14] J. L. McDonagh, A. Shkurti, D. J. Bray, R. L. Anderson, E. O. Pyzer-Knapp, Utilizing machine learning for efficient parameterization of coarse grained molecular force fields, Journal of Chemical Information and Modeling 59 (10) (2019) 4278–4288, pMID: 31549507. arXiv:https://doi.org/10.1021/acs.jcim.9b00646, doi:10.1021/acs.jcim.9b00646.
URL https://doi.org/10.1021/acs.jcim.9b00646

[15] A. L. Ferguson, Machine learning and data science in soft materials engineering, Journal of Physics: Condensed Matter 30 (4) (2017) 043002. doi:10.1088/1361-648x/aa98bd.

[16] Y. Xu, D. Verma, R. P. Sheridan, A. Liaw, J. Ma, N. M. Marshall, J. McIntosh, E. C. Sherer, V. Svetnik, J. M. Johnston, Deep dive into machine learning models for protein engineering, Journal of Chemical Information and Modeling 60 (6) (2020) 2773–2790, pMID: 32250622. arXiv:https://doi.org/10.1021/acs.jcim.0c00073, doi:10.1021/acs.jcim.0c00073.
URL https://doi.org/10.1021/acs.jcim.0c00073

[17] G. Klambauer, S. Hochreiter, M. Rarey, Machine learning in drug discovery, Journal of Chemical Information and Modeling 59 (3) (2019) 945–946. doi:10.1021/acs.jcim.9b00136.

[18] P. Geiger, C. Dellago, Neural networks for local structure detection in polymorphic systems, The Journal of Chemical Physics 139 (16) (2013) 164105. arXiv:https://doi.org/10.1063/1.4825111, doi:10.1063/1.4825111.
URL https://doi.org/10.1063/1.4825111

[19] E. Boattini, M. Dijkstra, L. Filion, Unsupervised learning for local structure detection in colloidal systems, The Journal of Chemical Physics 151 (15) (2019) 154901. doi:10.1063/1.5118867.

[20] T. Terao, A machine learning approach to analyze the structural formation of soft matter via image recognition, Soft Materials 18 (0) (2020) 215–227. doi:10.1080/1539445X.2020.1715433.

[21] K. Swanson, S. Trivedi, J. Lequieu, K. Swanson, R. Kondor, Deep learning for automated classification and characterization of amorphous materials, Soft Matter 16 (2020) 435–446. doi:10.1039/C9SM01903K.
URL http://dx.doi.org/10.1039/C9SM01903K

[22] W. Huang, P. Martin, H. L. Zhuang, Machine-learning phase prediction of high-entropy alloys, Acta Materialia 169 (2019) 225 – 236. doi:https://doi.org/10.1016/j.actamat.2019.03.012.
URL http://www.sciencedirect.com/science/article/pii/S1359645419301454

[23] S. Y. Lee, S. Byeon, H. S. Kim, H. Jin, S. Lee, Deep learning-based phase prediction of high-entropy alloys: Optimization, generation, and explanation, Materials & Design 197 (2021) 109260. doi:https://doi.org/10.1016/j.matdes.2020.109260.
URL http://www.sciencedirect.com/science/article/pii/S0264127520307954

[24] M. Chaplin, Anomalous properties of water, http://www.lsbu.ac.uk/water/anmlies.html (July 2020).

[25] R. Podgornik, Water and life: the unique properties of h2o, Journal of Biological Physics 37 (2011) 163–165. doi:10.1007/s10867-011-9217-9.
URL https://doi.org/10.1007/s10867-011-9217-9

[26] P. Gallo, K. Amann-Winkel, C. A. Angell, M. A. Anisimov, F. Caupin, C. Chakravarty, E. Lascaris, T. Loerting, A. Z. Panagiotopoulos, J. Russo, J. A. Sellberg, H. E. Stanley, H. Tanaka, C. Vega, L. Xu, L. G. M. Pettersson, Water: A tale of two liquids, Chemical Reviews 116 (13) (2016) 7463–7500, pMID: 27380438. arXiv:https://doi.org/10.1021/acs.chemrev.5b00750, doi:10.1021/acs.chemrev.5b00750.
URL https://doi.org/10.1021/acs.chemrev.5b00750

[27] J. Bachler, P. H. Handle, N. Giovambattista, T. Loerting, Glass polymorphism and liquid–liquid phase transition in aqueous solutions: experiments and computer simulations, Phys. Chem. Chem. Phys. 21 (2019) 23238–23268. doi:10.1039/C9CP02953B.
URL http://dx.doi.org/10.1039/C9CP02953B

[28] P. Lucas, S. Wei, C. A. Angell, Liquid-liquid phase transitions in glass-forming systems and their implications for memory technology, International Journal of Applied Glass Science 11 (2) (2020) 236–244. arXiv:https://ceramics.onlinelibrary.wiley.com/doi/pdf/10.1111/ijag.15109, doi:https://doi.org/10.1111/ijag.15109.
URL https://ceramics.onlinelibrary.wiley.com/doi/abs/10.1111/ijag.15109

[29] P. Poole, F. Sciortino, U. Essmann, H. Stanley, Phase-behavior of metastable water, Nature 360 (1992) 324–328. doi:10.1038/360324a0.

[30] P. H. Poole, R. K. Bowles, I. Saika-Voivod, F. Sciortino, Free energy surface of st2 water near the liquid-liquid phase transition, The Journal of Chemical Physics 138 (3) (2013) 034505. arXiv:https://doi.org/10.1063/1.

4775738, doi:10.1063/1.4775738.
URL https://doi.org/10.1063/1.4775738

[31] D. T. Limmer, D. Chandler, The putative liquid-liquid transition is a liquid-solid transition in atomistic models of water, The Journal of Chemical Physics 135 (13) (2011) 134503. doi:10.1063/1.3643333.

[32] D. T. Limmer, D. Chandler, The putative liquid-liquid transition is a liquid-solid transition in atomistic models of water. ii, The Journal of Chemical Physics 138 (21) (2013) 214504. arXiv:https://doi.org/10.1063/1.4807479, doi:10.1063/1.4807479.
URL https://doi.org/10.1063/1.4807479

[33] J. C. Palmer, R. Car, P. G. Debenedetti, The liquid–liquid transition in supercooled st2 water: a comparison between umbrella sampling and well-tempered metadynamics, Faraday Discuss. 167 (2013) 77–94. doi:10.1039/C3FD00074E.
URL http://dx.doi.org/10.1039/C3FD00074E

[34] J. C. Palmer, P. H. Poole, F. Sciortino, P. G. Debenedetti, Advances in computational studies of the liquid–liquid transition in water and water-like models, Chemical Reviews 118 (18) (2018) 9129–9151, pMID: 30152693. arXiv:https://doi.org/10.1021/acs.chemrev.8b00228, doi:10.1021/acs.chemrev.8b00228.
URL https://doi.org/10.1021/acs.chemrev.8b00228

[35] H. Stanley, L. Cruz, S. Harrington, P. Poole, S. Sastry, F. Sciortino, F. Starr, R. Zhang, Cooperative molecular motions in water: The liquid-liquid critical point hypothesis, Physica A: Statistical Mechanics and its Applications 236 (1) (1997) 19 – 37, proceedings of the Workshop on Current Problems in Complex Fluids. doi:https://doi.org/10.1016/S0378-4371(96)00429-3.
URL http://www.sciencedirect.com/science/article/pii/S0378437196004293

[36] O. Mishima, H. Stanley, The relationship between liquid, supercooled and glassy water, Nature 396 (11 1998). doi:10.1038/24540.

[37] H. Stanley, S. Buldyrev, O. Mishima, M. Sadr-Lahijany, A. Scala, F. Starr, Unsolved mysteries of water in its liquid and glassy phases, Journal of Physics: Condensed Matter 12 (2000) A403. doi:10.1088/0953-8984/12/8A/355.

[38] F. Sciortino, E. la Nave, P. Tartaglia, Physics of the liquid-liquid critical point, Physical review letters 91 (2003) 155701. doi:10.1103/PhysRevLett.91.155701.

[39] P. G. Debenedetti, Supercooled and glassy water, Journal of Physics: Condensed Matter 15 (45) (2003) R1669–R1726. doi:10.1088/0953-8984/

15/45/r01.
URL https://doi.org/10.1088%2F0953-8984%2F15%2F45%2Fr01

[40] P. H. Handle, T. Loerting, F. Sciortino, Supercooled and glassy water: Metastable liquid(s), amorphous solid(s), and a no-man's land, Proceedings of the National Academy of Sciences 114 (51) (2017) 13336–13344. arXiv:https://www.pnas.org/content/114/51/13336.full.pdf, doi:10.1073/pnas.1700103114.
URL https://www.pnas.org/content/114/51/13336

[41] K. Amann-Winkel, C. Gainaru, P. H. Handle, M. Seidl, H. Nelson, R. Böhmer, T. Loerting, Water's second glass transition, Proceedings of the National Academy of Sciences 110 (44) (2013) 17720–17725. arXiv:https://www.pnas.org/content/110/44/17720.full.pdf, doi:10.1073/pnas.1311718110.
URL https://www.pnas.org/content/110/44/17720

[42] A. Taschin, P. Bartolini, R. Eramo, R. Righini, R. Torre, Evidence of two distinct local structures of water from ambient to supercooled conditions, Nat. Comm. 4 (2013) 2401.

[43] K. H. Kim, A. Späh, H. Pathak, F. Perakis, D. Mariedahl, K. Amann-Winkel, J. A. Sellberg, J. H. Lee, S. Kim, J. Park, K. H. Nam, T. Katayama, A. Nilsson, Maxima in the thermodynamic response and correlation functions of deeply supercooled water, Science 358 (6370) (2017) 1589–1593. arXiv:https://science.sciencemag.org/content/358/6370/1589.full.pdf, doi:10.1126/science.aap8269.
URL https://science.sciencemag.org/content/358/6370/1589

[44] F. Caupin, Escaping the no man's land: Recent experiments on metastable liquid water, Journal of Non-Crystalline Solids 407 (2015) 441 – 448, 7th IDMRCS: Relaxation in Complex Systems. doi:https://doi.org/10.1016/j.jnoncrysol.2014.09.037.
URL http://www.sciencedirect.com/science/article/pii/S002230931400492X

[45] N. J. Hestand, J. L. Skinner, Perspective: Crossing the widom line in no man's land: Experiments, simulations, and the location of the liquid-liquid critical point in supercooled water, The Journal of Chemical Physics 149 (14) (2018) 140901. arXiv:https://doi.org/10.1063/1.5046687, doi:10.1063/1.5046687.
URL https://doi.org/10.1063/1.5046687

[46] K. H. Kim, K. Amann-Winkel, N. Giovambattista, A. Späh, F. Perakis, H. Pathak, M. L. Parada, C. Yang, D. Mariedahl, T. Eklund, T. J. Lane, S. You, S. Jeong, M. Weston, J. H. Lee, I. Eom, M. Kim, J. Park, S. H. Chun, P. H. Poole, A. Nilsson, Experimental observation of the liquid-liquid transition in bulk supercooled water under pressure, Science

370 (6519) (2020) 978–982. arXiv:https://science.sciencemag.org/content/370/6519/978.full.pdf, doi:10.1126/science.abb9385.
URL https://science.sciencemag.org/content/370/6519/978

[47] C. G. Salzmann, Advances in the experimental exploration of water's phase diagram, The Journal of Chemical Physics 150 (6) (2019) 060901. arXiv:https://doi.org/10.1063/1.5085163, doi:10.1063/1.5085163.
URL https://doi.org/10.1063/1.5085163

[48] P. G. Debenedetti, F. Sciortino, G. H. Zerze, Second critical point in two realistic models of water, Science 369 (6501) (2020) 289–292. arXiv:https://science.sciencemag.org/content/369/6501/289.full.pdf, doi:10.1126/science.abb9796.
URL https://science.sciencemag.org/content/369/6501/289

[49] F. Martelli, F. Leoni, F. Sciortino, J. Russo, Connection between liquid and non-crystalline solid phases in water, The Journal of Chemical Physics 153 (10) (2020) 104503. arXiv:https://doi.org/10.1063/5.0018923, doi:10.1063/5.0018923.
URL https://doi.org/10.1063/5.0018923

[50] T. E. Gartner, L. Zhang, P. M. Piaggi, R. Car, A. Z. Panagiotopoulos, P. G. Debenedetti, Signatures of a liquid–liquid transition in an ab initio deep neural network model for water, Proceedings of the National Academy of Sciences 117 (42) (2020) 26040–26046. arXiv:https://www.pnas.org/content/117/42/26040.full.pdf, doi:10.1073/pnas.2015440117.
URL https://www.pnas.org/content/117/42/26040

[51] B. Monserrat, J. Brandenburg, E. Engel, B. Cheng, Liquid water contains the building blocks of diverse ice phases, Nature Communications 11 (2020) 5757. doi:10.1038/s41467-020-19606-y.
URL https://doi.org/10.1038/s41467-020-19606-y

[52] M. Fulford, M. Salvalaglio, C. Molteni, Deepice: A deep neural network approach to identify ice and water molecules, Journal of Chemical Information and Modeling 59 (5) (2019) 2141–2149, pMID: 30875217. arXiv:https://doi.org/10.1021/acs.jcim.9b00005, doi:10.1021/acs.jcim.9b00005.
URL https://doi.org/10.1021/acs.jcim.9b00005

[53] S. S. Schoenholz, Combining machine learning and physics to understand glassy systems, Journal of Physics: Conference Series 1036 (2018) 012021. doi:10.1088/1742-6596/1036/1/012021.
URL https://doi.org/10.1088/1742-6596/1036/1/012021

[54] E. Boattini, S. Marín-Aguilar, S. Mitra, G. Foffi, F. Smallenburg, L. Filion, Autonomously revealing hidden local structures in supercooled liquids, Nature Communications 11 (2020) 5479. arXiv:https://doi.org/10.1063/1.3463424, doi:10.1038/s41467-020-19286-8.
URL https://doi.org/10.1038/s41467-020-19286-8

[55] Z. Zhao, C. A. Angell, Apparent first-order liquid–liquid transition with pre-transition density anomaly, in water-rich ideal solutions, Angewandte Chemie International Edition 55 (7) (2016) 2474–2477. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.201510717, doi:https://doi.org/10.1002/anie.201510717.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.201510717

[56] V. Champreda, D. Stuckey, A. Boontawan, Separation of methanol/water mixtures from dilute aqueous solutions using pervaporation technique, in: Advances in Chemical Engineering II, Vol. 550 of Advanced Materials Research, Trans Tech Publications Ltd, 2012, pp. 3004–3007. doi:10.4028/www.scientific.net/AMR.550-553.3004.

[57] C. R. Smith, Alcohol as a disinfectant against the tubercle bacillus, Public Health Reports (1896-1970) 62 (36) (1947) 1285–1295.
URL http://www.jstor.org/stable/4586265

[58] T. V. N. Nguyen, L. Paugam, P. Rabiller, M. Rabiller-Baudry, Study of transfer of alcohol (methanol, ethanol, isopropanol) during nanofiltration in water/alcohol mixtures, Journal of Membrane Science 601 (2020) 117907. doi:https://doi.org/10.1016/j.memsci.2020.117907.
URL http://www.sciencedirect.com/science/article/pii/S0376738819324639

[59] N. K. Hermkens, R. L. Aspers, M. C. Feiters, F. P. Rutjes, M. Tessari, Trace analysis in water-alcohol mixtures by continuous p-h2 hyperpolarization at high magnetic field, Magnetic Resonance in Chemistry 56 (7) (2018) 633–640. doi:https://doi.org/10.1002/mrc.4692.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/mrc.4692

[60] G. Pálinkás, E. Hawlicka, K. Heinzinger, Molecular dynamics simulations of water-methanol mixtures, Chemical Physics 158 (1) (1991) 65 – 76. doi:https://doi.org/10.1016/0301-0104(91)87055-Z.
URL http://www.sciencedirect.com/science/article/pii/0301010491187055Z

[61] D. González-Salgado, I. Nezbeda, Excess properties of aqueous mixtures of methanol: Simulation versus experiment, Fluid Phase Equilibria 240 (2) (2006) 161 – 166. doi:https://doi.org/10.1016/j.fluid.2005.12.007.
URL http://www.sciencedirect.com/science/article/pii/S0378381205004966

[62] D. Corradini, Z. Su, H. E. Stanley, P. Gallo, A molecular dynamics study of the equation of state and the structure of supercooled aqueous solutions of methanol, The Journal of Chemical Physics 137 (18) (2012) 184503. arXiv:https://doi.org/10.1063/1.4767060, doi:10.1063/1.4767060.
URL https://doi.org/10.1063/1.4767060

[63] G. Munaò, T. Urbic, Structure and thermodynamics of core-softened models for alcohols, The Journal of Chemical Physics 142 (21) (2015) 214508. arXiv:https://doi.org/10.1063/1.4922164, doi:10.1063/1.4922164.
URL https://doi.org/10.1063/1.4922164

[64] D. González-Salgado, K. Zemánková, E. G. Noya, E. Lomba, Temperature of maximum density and excess thermodynamics of aqueous mixtures of methanol, The Journal of Chemical Physics 144 (18) (2016) 184505. arXiv:https://doi.org/10.1063/1.4948611, doi:10.1063/1.4948611.
URL https://doi.org/10.1063/1.4948611

[65] A. P. Furlan, E. Lomba, M. C. Barbosa, Temperature of maximum density and excess properties of short-chain alcohol aqueous solutions: A simplified model simulation study, The Journal of Chemical Physics 146 (14) (2017) 144503. arXiv:https://doi.org/10.1063/1.4979806, doi:10.1063/1.4979806.
URL https://doi.org/10.1063/1.4979806

[66] D. González-Salgado, J. Troncoso, E. Lomba, The temperature of maximum density for amino acid aqueous solutions. an experimental and molecular dynamics study, Fluid Phase Equilibria 521 (2020) 112703. doi:https://doi.org/10.1016/j.fluid.2020.112703.
URL http://www.sciencedirect.com/science/article/pii/S0378381220302491

[67] M. S. Marques, V. F. Hernandes, E. Lomba, J. R. Bordin, Competing interactions near the liquid-liquid phase transition of core-softened water/methanol mixtures, Journal of Molecular Liquids 320 (2020) 114420. doi:https://doi.org/10.1016/j.molliq.2020.114420.
URL http://www.sciencedirect.com/science/article/pii/S0167732220354945

[68] M. S. Marques, V. F. Hernandes, J. R. Bordin, Core-softened water-alcohol mixtures: the solute-size effects, arXiv:2102.09485.
URL https://arxiv.org/abs/2102.09485

[69] G. Franzese, Differences between discontinuous and continuous soft-core attractive potentials: The appearance of density anomaly, Journal of Molecular Liquids 136 (3) (2007) 267 – 273, eMLG/JMLG 2006. doi:https://doi.org/10.1016/j.molliq.2007.08.021.
URL http://www.sciencedirect.com/science/article/pii/S016773220700150X

[70] P. Vilaseca, G. Franzese, Isotropic soft-core potentials with two characteristic length scales and anomalous behaviour, Journal of Non-Crystalline Solids 357 (2011) 419–426. doi:10.1016/j.jnoncrysol.2010.07.053.

[71] E. A. Jagla, Core-softened potentials and the anomalous properties of water, The Journal of Chemical Physics 111 (19) (1999) 8980–8986. arXiv:

https://doi.org/10.1063/1.480241, doi:10.1063/1.480241.
URL https://doi.org/10.1063/1.480241

[72] A. Oliveira, P. Netz, M. Barbosa, Which mechanism underlies the water-like anomalies in core-softened potentials?, The European Physical Journal B 64 (2008) 481–486. doi:10.1140/epjb/e2008-00101-6.

[73] Y. D. Fomin, E. N. Tsiok, V. N. Ryzhov, Inversion of sequence of diffusion and density anomalies in core-softened systems, J. Chem. Phys. 135 (2011) 234502.

[74] J. Bordin, M. Barbosa, Flow and structure of fluids in functionalized nanopores, Physica A: Statistical Mechanics and its Applications 467 (10 2016). doi:10.1016/j.physa.2016.10.007.

[75] G. Munaò, T. Urbic, Structure and thermodynamics of core-softened models for alcohols, The Journal of Chemical Physics 142 (21) (2015) 214508. arXiv:https://doi.org/10.1063/1.4922164, doi:10.1063/1.4922164.
URL https://doi.org/10.1063/1.4922164

[76] P. J. Steinhardt, D. R. Nelson, M. Ronchetti, Bond-orientational order in liquids and glasses, Phys. Rev. B 28 (1983) 784–805. doi:10.1103/PhysRevB.28.784.
URL https://link.aps.org/doi/10.1103/PhysRevB.28.784

[77] W. Lechner, C. Dellago, Accurate determination of crystal structures based on averaged local bond order parameters, The Journal of Chemical Physics 129 (11) (2008) 114707. arXiv:https://doi.org/10.1063/1.2977970, doi:10.1063/1.2977970.
URL https://doi.org/10.1063/1.2977970

[78] E. Boattini, M. Ram, F. Smallenburg, L. Filion, Neural-network-based order parameters for classification of binary hard-sphere crystal structures, Molecular Physics 116 (21-22) (2018) 3066–3075. arXiv:https://doi.org/10.1080/00268976.2018.1483537, doi:10.1080/00268976.2018.1483537.
URL https://doi.org/10.1080/00268976.2018.1483537

[79] M. Allen, D. Tildesley, D. Tildesley, Computer Simulation of Liquids, Oxford University Press, 2017.
URL https://books.google.com.br/books?id=nlExDwAAQBAJ

[80] V. Ramasubramani, B. D. Dice, E. S. Harper, M. P. Spellings, J. A. Anderson, S. C. Glotzer, freud: A software suite for high throughput analysis of particle simulation data, Computer Physics Communications 254 (2020) 107275. doi:https://doi.org/10.1016/j.cpc.2020.107275.
URL http://www.sciencedirect.com/science/article/pii/S0010465520300916

[81] F. Chollet, et al., Keras, https://keras.io (2015).

[82] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL https://www.tensorflow.org/

[83] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Y. W. Teh, M. Titterington (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Vol. 9 of Proceedings of Machine Learning Research, JMLR Workshop and Conference Proceedings, Chia Laguna Resort, Sardinia, Italy, 2010, pp. 249–256.
URL http://proceedings.mlr.press/v9/glorot10a.html

[84] D. P. Kingma, J. L. Ba, Adam: A method for stochastic gradient descent, in: ICLR: International Conference on Learning Representations, 2015, pp. 1–15.

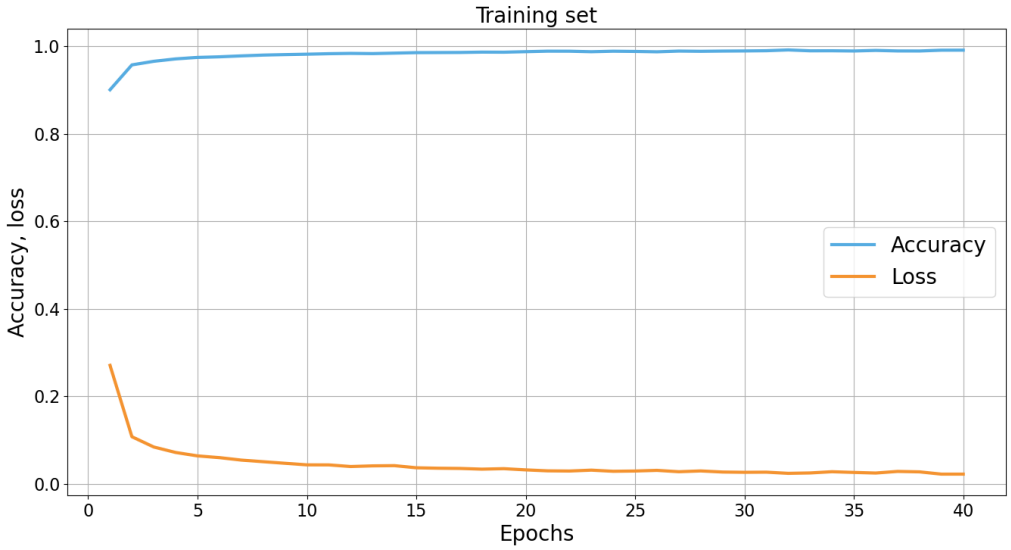
[85] V. Holten, C. E. Bertrand, M. A. Anisimov, J. V. Sengers, Thermodynamics of supercooled water, The Journal of Chemical Physics 136 (9) (2012) 094507. arXiv:https://doi.org/10.1063/1.3690497, doi:10.1063/1.3690497.
URL https://doi.org/10.1063/1.3690497

[86] V. Bianco, G. Franzese, Hydrogen bond correlated percolation in a supercooled water monolayer as a hallmark of the critical region, Journal of Molecular Liquids 285 (2019) 727–739. doi:https://doi.org/10.1016/j.molliq.2019.04.090.
URL https://www.sciencedirect.com/science/article/pii/S0167732218350086

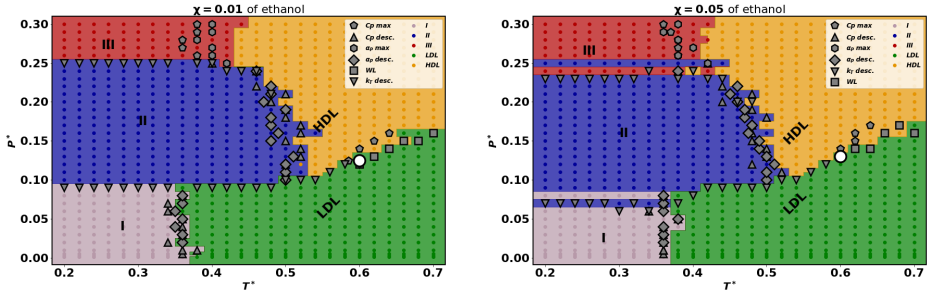
[87] W. P. Krekelberg, J. Mittal, V. Ganesan, T. M. Truskett, Structural anomalies of fluids: Origins in second and higher coordination shells, Phys. Rev. E 77 (2008) 041201. doi:10.1103/PhysRevE.77.041201.
URL https://link.aps.org/doi/10.1103/PhysRevE.77.041201

[88] P. Vilaseca, G. Franzese, Softness dependence of the anomalies for the continuous shouldered well potential, The Journal of Chemical Physics 133 (8) (2010) 084507. arXiv:https://doi.org/10.1063/1.3463424, doi:10.1063/1.3463424.
URL https://doi.org/10.1063/1.3463424

6. Supplementary Material



(a)



(b) (c)

Figure 7: (a) Training accuracy and loss. (b) Phase diagrams of water-ethanol mixture with concentration $\chi = 0.01$ and (c) $\chi = 0.05$, predicted by the neural network.

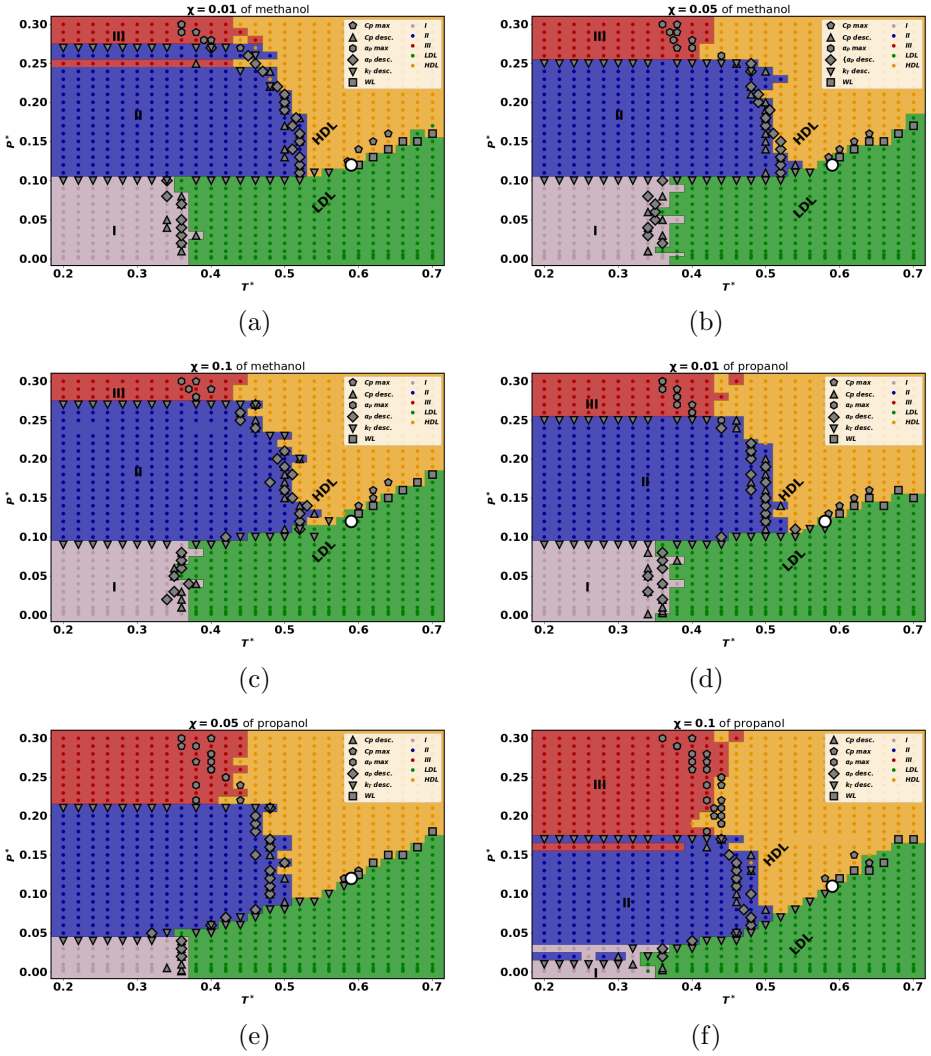


Figure 8: Phase diagrams of water-methanol mixture with concentration (a) $\chi = 0.01$, (b) $\chi = 0.05$ and (c) $\chi = 0.1$ and water-propanol mixture with concentration (d) $\chi = 0.01$, (e) $\chi = 0.05$ and (f) $\chi = 0.1$, predicted by the neural network.

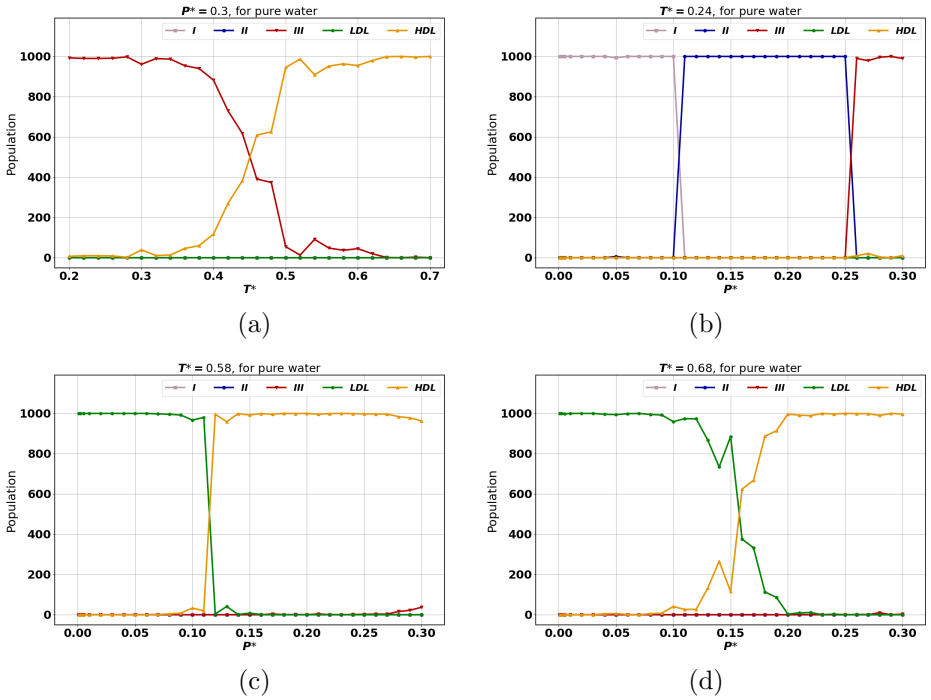


Figure 9: Population of pure water (a) as a function of temperature for fixed pressure equal to 0.3, and as a function of pressure for fixed temperature equal to (b) 0.24, (c) 0.58 and (d) 0.68.

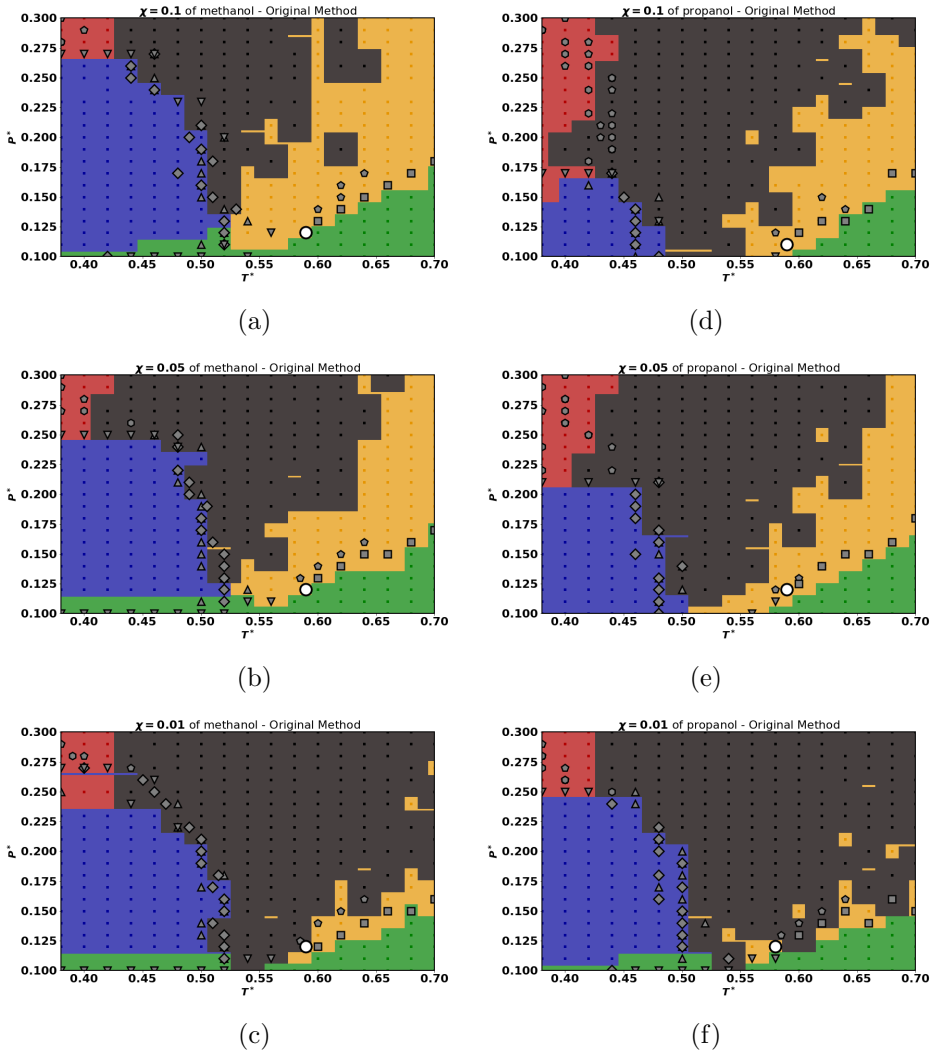


Figure 10: Phase diagrams of water-methanol mixture with concentration (a) $\chi = 0.01$, (b) $\chi = 0.05$ and (c) $\chi = 0.1$ and water-propanol mixture with concentration (d) $\chi = 0.01$, (e) $\chi = 0.05$ and (f) $\chi = 0.1$, predicted by the neural network, for $0.38 \leq \mathbf{T}^* \leq 0.70$ and $0.10 \leq \mathbf{P}^* \leq 0.30$, with the metastable phase explicit.