UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Rio Claro

Programa de Pós-Graduação em Ciência da Computação

Leonardo Souza Campos

Investigação de redes neurais convolucionais na geração de modelos de substituição em processo de desenvolvimento de peças industriais

Rio Claro
2024

UNIVERSIDADE ESTADUAL PAULISTA
“Júlio de Mesquita Filho”
Instituto de Geociências e Ciências Exatas
Câmpus de Rio Claro

Leonardo Souza Campos

Investigação de redes neurais convolucionais na geração de modelos de substituição em processo de desenvolvimento de peças industriais

Dissertação de Mestrado apresentada ao Instituto de Geociências e Ciências Exatas do Campus de Rio Claro, da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Denis Henrique Pinheiro Salvadeo

Rio Claro - SP
2024

S729i

Souza Campos, Leonardo
Investigação de redes neurais convolucionais na geração de modelos de substituição em processo de desenvolvimento de peças industriais / Leonardo Souza Campos. -- Rio Claro, 2024
97 p. : tabs., fotos

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP), Instituto de Geociências e Ciências Exatas, Rio Claro
Orientador: Denis Henrique Pinheiro Salvadeo

1. Redes neurais (Computação). 2. Dinâmica dos fluidos computacional. 3. Dinâmica dos fluidos. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

UNIVERSIDADE ESTADUAL PAULISTA
"Júlio de Mesquita Filho"
Instituto de Geociências e Ciências Exatas
Câmpus de Rio Claro

LEONARDO SOUZA
CAMPOS

Investigação de redes neurais convolucionais na geração de modelos de substituição em processo de desenvolvimento de peças industriais

Dissertação de Mestrado apresentada ao Instituto de Geociências e Ciências Exatas do Câmpus de Rio Claro, da Universidade Estadual Paulista "Júlio de Mesquita Filho", como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Comissão Examinadora:

Prof. Dr. DENIS HENRIQUE PINHEIRO SALVADEO
IGCE / UNESP/Rio Claro (SP)

Prof. Dr. FABRÍCIO APARECIDO BREVE
IGCE / UNESP/Rio Claro (SP)

Prof. Dr. DIEGO SAMUEL RODRIGUES
FT / UNICAMP/Campinas (SP)

Conceito: Aprovado.

Rio Claro (SP), 05 de Agosto de 2024

AGRADECIMENTOS

Gostaria de agradecer ao amparo que recebi para a realização desta dissertação. Inicialmente, gostaria de agradecer a orientação do Prof. Dr. Denis Henrique Pinheiro Salvadeo, que foi fundamental para a coordenação, acompanhamento e recomendações técnicas para a execução do trabalho.

Agradeço também à *Whirlpool Latin America*, por ter disposto de recursos tecnológicos e humanos, que foram essenciais para a obtenção dos dados e realização deste projeto. Agradecimento especial aos colegas João Curiel e Mathiu Barrezueta, que lideraram os aspectos técnicos e geração do banco de dados das simulações de fluidodinâmica computacional. Agradeço também a meu gestor, Aliander Filgueiras da Silva, pela disponibilidade de tempo e recursos que permitiram a execução deste projeto em meio às demais demandas da companhia.

Por último, e mais importante, agradeço o apoio de minha família, meu pai Marcelo, minha mãe Adriana, meu irmão Rodrigo e minha esposa Flávia, que foram essenciais por me prover suporte emocional e apoio durante toda a elaboração da dissertação.

RESUMO

As ferramentas de aprendizado de máquina, em especial redes neurais artificiais, têm trazido enormes avanços em diversas áreas da ciência e tecnologia. Um dos campos em que o aprendizado profundo tem potencial para trazer ganhos significativos é sua utilização em conjunto com ferramentas de simulação computacional, tais como fluidodinâmica computacional. Este trabalho estudou a aplicação de arquiteturas de redes neurais artificiais, especificamente redes convolucionais, na geração de um modelo de substituição que se utiliza de técnicas de aprendizado supervisionado no treinamento de um modelo capaz de representar o escoamento e a interação fluido-estrutura de um determinado perfil fluidodinâmico. Este modelo foi treinado a partir de resultados de simulação obtidos por dados de simulação de fluidodinâmica computacional de dispensadores de sabão em pó de lavadoras de roupas, relacionado aos campos escalares de cisalhamento e fração volumétrica na superfície crítica do componente. Os hiperparâmetros ótimos da rede neural foram obtidos através do uso de experimentos fatoriais para ambos os campos escalares. Um modelo de regressão também foi criado para obter o efeito de cada variável geométrica do modelo paramétrico utilizado para gerar os indivíduos de treinamento com o erro da rede. Os resultados obtidos demonstram boa aderência entre os resultados obtidos com o modelo de substituição com os resultados obtidos via simulação. Vale destacar também que a arquitetura de rede neural e os algoritmos de otimização e treinamento podem ser aplicados a qualquer outro domínio físico, não sendo limitado ao utilizado para treinamento neste estudo.

Palavras-chave: Aprendizado de máquina, aprendizado profundo, redes neurais convolucionais, fluidodinâmica computacional, modelos de substituição, desenvolvimento de peças.

ABSTRACT

Machine learning tools, especially artificial neural networks, have brought enormous advances in several areas of science and technology. One of the fields in which deep learning has the potential to bring significant gains is its use in conjunction with computational simulation tools, such as computational fluid dynamics. This work studied the application of artificial neural network architectures, specifically convolutional networks, in the generation of a surrogate model that applies supervised learning techniques to train a model capable of representing the flow and fluid-structure interaction of a given fluid dynamic profile. This model was trained from simulation results obtained by computational fluid dynamics simulation data of powder dispensers utilized in wash machines, related to the scalar fields of shear and volume fraction in the critical surface of the component. The optimal hyperparameters of the neural network were obtained through the use of factorial experiments for both scalar fields. A regression model was also created to obtain the effect of each geometric variable of the parametric model used to generate the training individuals on the network training error. The obtained results demonstrate good adherence between the results from the surrogate model and the results obtained via simulation. It is also worth highlighting that the neural network architecture and the optimization and training algorithms can be applied to any other physical domain, not being limited to the one used in this study.

Keywords: Machine learning, deep learning, convolutional neural networks, computational fluid dynamics, surrogate models, product development.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

Dentro do âmbito da engenharia, especificamente a engenharia de produto, há uma tendência crescente de se aplicar técnicas de otimização e design proativo (HARRIES, 2020). Há diversos fatores que fortalecem esta prática, cerceados principalmente nas necessidades do mercado, que devem ser contempladas na concepção do projeto de um produto.

Considerando-se a concorrência do mercado em determinados setores e dada a evolução e melhoria histórica dos produtos relacionados a estes setores, tais como bens e serviços, setor aeronáutico, setor automobilístico, setor de energia, entre outros, projetos de desenvolvimento de produto devem contemplar, a fim de apresentar um plano de negócio saudável, um balanço entre preço, qualidade, margem de lucro, tempo de execução, risco técnico, entre diversas outras variáveis.

Desta maneira, técnicas de otimização tem papel fundamental neste cenário, tendo em vista que contribuem em diversas características (HARRIES, 2020):

a) Melhoria da performance do produto sem a necessidade de prototipagens sucessivas;
b) Estimativa da performance do produto antes de sua confecção ou prototipagem;
c) Redução dos custos relacionados à confecção de protótipos, danos gerados por testes sem sucesso, etc;
d) Redução do cronograma de implementação de um projeto, dados os ganhos mencionados anteriormente.

Uma das principais ferramentas que vêm sendo aplicadas com muito sucesso na etapa de otimização e concepção de produto são as simulações computacionais, sendo duas entre as principais técnicas: a) análise estrutural por elementos finitos (ALVES FILHO, 2018) e b) fluidodinâmica computacional por volumes finitos (MALALASEKERA,1995).

Em suma, ambas as técnicas tratam-se de algoritmos computacionais, cujo princípio de funcionamento é baseado na discretização de um meio contínuo complexo, para os quais a solução analítica é extremamente complexa ou inexistente, em domínios menores, cujas equações governantes são mais simples, e que podem ser resolvidas como um sistema de equações (ALVES FILHO, 2018).

A técnica de fluidodinâmica computacional (CFD) (MALALASEKERA,1995) se baseia na resolução das equações de transporte, especialmente balanço de massa, momento e energia, para obter os campos de pressão, velocidade e as forças geradas por determinada interação fluido-estrutura.

A área de fenômenos de transporte faz uso considerável das ferramentas de análise de fluidodinâmica computacional em função de grande parte das equações diferenciais que regem o escoamento de fluidos não possuirem soluções analíticas, sendo os modelos de discretização computacional aliados a modelos de turbulência a única alternativa de modelamento destes problemas (HARRIES, 2020).

Ainda que as simulações fluidodinâmicas representem um avanço significativo na área do design proativo e otimização, elas ainda possuem restrições à aplicação em conjunto com algoritmos de otimização tanto heurísticos, quanto esquemas baseados em gradiente (BOUHLEL, 2020).

Isto se deve ao fato de que, mesmo domínios de complexidade média, necessitam de horas ou até dias de execução, mesmo considerando-se alta capacidade computacional para cada modelo simulado. Desta maneira, métodos de otimização que necessitam de diversas iterações e modelos sequenciais de treinamento tornam-se inviáveis, pois o tempo de geração da população de treinamento se torna por vezes impeditivo em relação ao cronograma de execução do projeto (LALONDE et al., 2021).

Com o propósito de vencer esta limitação e aliado ao fortalecimento nas últimas décadas da aplicação de técnicas de aprendizado profundo (*deep learning)* (DU et al., 2021) em diversas áreas com resultados excepcionais, uma das linhas de pesquisas atualmente recorrentes relacionadas à fluidodinâmica computacional é a geração de modelos, principalmente baseados em arquiteturas de redes neurais profundas com função de ativação da camada de saída linear.

O objetivo é ser capaz de utilizar resultados de simulação computacional, especificamente os valores de cada elemento do campo discretizado de pressão ou velocidade, como dados de entrada para treinamento, visando gerar um modelo capaz de prever a performance fluidodinâmica e as variáveis pertinentes a cada área, de maneira a tornar possível a estimativa de performance de uma nova proposta sem a necessidade de execução da simulação fluidodinâmica. Estes modelos são denominados modelos de substituição, ou *surrogate models* (QIAN et al., 2006).

O primeiro grupo de técnicas de modelos de substituição se baseia em utilizar as redes neurais como um regressor direto das variáveis de performance pertinentes à determinada aplicação, como o coeficiente de arrasto e sustentação de um aerofólio ou a potência de um rotor de turbina. As entradas para este regressor usualmente são as variáveis geométricas e variáveis físicas relacionadas ao problema, tais como velocidade, temperatura, rugosidade, etc. A camada de saída destes modelos tipicamente constitui um ou mais neurônios com saída linear, representando as variáveis de performance que se deseja estimar. Os dados de treinamento do modelo são tipicamente as variáveis de desempenho estimadas utilizando a simulação de CFD. Usualmente neste grupo de trabalho se usam Multilayer Perceptrons (MLPs) (ROSENBLATT, 1961).

Em continuação aos modelos de substituição baseados apenas na variável de desempenho de resposta, esforços subsequentes vêm sendo focados em utilizar aprendizado profundo na estimativa não apenas destas variáveis, mas também dos campos de pressão, temperatura e velocidade, entre outros. Este tipo de arquitetura, além de trazer melhores resultados na previsão das variáveis, inclusive em casos limítrofes, onde as MLPs tradicionais falham, permite uma análise mais aprofundada do comportamento físico do sistema, o que traz vantagem na interpretação dos resultados gerados e verificação da sanidade do modelo. Estes modelos utilizam arquiteturas de redes convolucionais (LECUN, 1998) de modo a lidar com o grande volume de dados e a melhor capacidade de trabalhar com dados estruturados (WANG et al., 2021; QIWAN et al., 2022).

Outra linha de pesquisa na qual esforços vêm sendo concentrados que relaciona aprendizado profundo e CFD são modelos geracionais. Assim como mencionado, modelos de redes neurais necessitam de populações razoavelmente grandes para apresentar resultados aceitáveis, que muitas vezes é uma limitação até pela falta de exemplares físicos para geração do banco de dados. Deste modo, redes neurais adversariais generativas (GANs) (GOODFELLOW et al., 2014) vêm sendo aplicadas no intuito de propiciar a geração de uma população de treinamento suficientemente similar à população real, de maneira a melhorar o próprio treinamento dos modelos de substituição, similar à técnica de *image augmentation* (PEDRINI et al., 2008) aplicada a modelos de treinamento de classificação por imagem.

O intuito deste projeto de mestrado foi baseado na geração de um modelo de substituição para a simulação fluidodinâmica a partir de um banco de dados gerado considerando uma geometria parametrizada de um componente base, definida a partir de um conjunto de variáveis independentes discretas. A aplicação para qual se pretendeu realizar o modelo de substituição está relacionada ao setor de bens e serviços, mais especificamente voltado para o setor de bens de consumo, em aplicação relacionada a lavadoras de roupas automáticas.

Um dos principais componentes de uma lavadora automática de roupas é o dispensador de insumos, responsável pela migração dos insumos de lavagem (sabão, amaciante e alvejante) para a roupa que se pretende lavar no momento adequado. Dentre os tipos principais de dispensador de insumos, o mais comum é o dispensador de gaveta.

Por mais que haja boas práticas quanto a seu dimensionamento, o sistema é afetado por diversas variáveis de ruído provenientes do uso do consumidor, tais como temperatura e vazão de água, inclinação do produto, tipo e quantidade de insumos, além das próprias variáveis de projeto. Sendo assim, a cada novo desenvolvimento de produto onde há necessidade de desenvolvimento de um novo sistema de dispensamento, as ferramentas de CFD são utilizadas na etapa de concepção do modelo, além de estudos de correlação com todos os testes experimentais.

O escopo deste projeto envolveu, especificamente, utilizar as técnicas de aprendizado de máquina durante a etapa de desenvolvimento do componente. Por mais que as técnicas de CFD tenham representado um avanço significativo no desenvolvimento de produto, o tempo de execução da simulação a torna impeditiva para que um processo de otimização ocorra, no qual diversas iterações diferentes seriam executadas e um refinamento sucessivo da geometria em questão ocorresse. Este tempo de execução não está apenas relacionado ao tempo de execução da simulação, mas também aos fluxos processuais da companhia, relacionado a priorização de recursos humanos e físicos para a execução da simulação, além do próprio trabalho de pré-processamento da geometria, em um processo conhecido em simulação como geração da malha.

Desta maneira, a execução de um único tratamento de simulação pode levar semanas, sendo grande parte deste tempo consumido por priorização de recursos,

burocracias internas, além do próprio trabalho de pré-processamento, que é necessário para cada peça individual.

Neste sentido, a obtenção de um modelo de substituição que seja suficientemente equivalente a uma simulação de CFD conseguiria prover resultados similares em um tempo muitas vezes menor, em questão de minutos. Além de prover agilidade ao processo de desenvolvimento de produto, tal técnica permitiria a aplicação de métodos de otimização, tanto baseado em gradiente quanto heurísticos. Isto permitiria a obtenção de mínimos globais específicos para as restrições de cada projeto, tornando o modelo de substituição flexível a aplicações variadas.

Em outras palavras, cada projeto em si apresenta um certo conjunto de restrições ao engenheiro, sejam elas restrições geométricas, como altura e largura máximas que o sistema pode ocupar, como funcionais, tal como a faixa de vazão e temperatura de água que o sistema deve operar. Tendo em mãos um modelo de substituição como comentado acima, o engenheiro seria capaz de encontrar uma combinação das demais variáveis geométricas considerando-se toda a combinação de restrições que lhe é imputada e encontrar o ponto ótimo de operação para aquela condição específica. O Capítulo 2 trará detalhes mais específicos a respeito do funcionamento do componente e sobre a proposta de trabalho.

Além da concepção do modelo de rede neural em si, o trabalho aplicou técnicas de experimentação baseada em experimentos fatoriais para dois contextos diferentes.

Primeiramente, os dados de treinamento foram gerados tendo como base um experimento fatorial fracionado, contendo geometrias geradas a partir de combinações das variáveis independentes mais críticas à aplicação. Esta aplicação permite uma análise de regressão baseada em entender como o nível de cada variável independente afeta o grau de erro esperado na estimativa do modelo. Esta estimativa de erro é uma informação complementar ao engenheiro a respeito do grau de erro esperado do modelo de substituição, além de indicar regiões onde o erro é mais elevado e deve ser endereçado em novas amostragens.

Posteriormente, técnicas de experimentação foram utilizadas para encontrar a combinação de hiperparâmetros de treinamento de rede neural que minimizem o erro. Este processo foi feito tanto para o campo escalar de fração volumétrica *(VOF)*

quanto de cisalhamento *(shear)*, que são duas variáveis críticas para a aplicação em questão, o que será abordado em capítulos subsequentes desta dissertação.

A principal contribuição relacionada ao trabalho desenvolvido é oferecer um modelo de substituição que, conforme mencionado anteriormente, tenha a capacidade de gerar resultados satisfatórios quando comparados à simulação computacional de fluidodinâmica. Conforme mencionado anteriormente, durante a cadência de um projeto de desenvolvimento de produto, a quantidade de demanda para a equipe de simulação geralmente é maior que a quantidade de recursos disponíveis para sua execução. Desta maneira, por vezes o tempo para execução da simulação abrange em grande parte o tempo de espera e priorização de demandas. Além do mais, usualmente a natureza de simulações que se executa em grande parte no projeto são muito similares às outras já executadas em projetos anteriores, porém necessárias de serem executadas. Tendo em mãos um modelo de substituição deste gênero, este pode ser utilizado para agilizar este tipo de demanda, agilizando consideravelmente os resultados.

Além do mais, a arquitetura de rede neural proposta, apesar de utilizada especificamente para a aplicação de um dispensador automático de insumos, pode ser estendida para qualquer outro domínio computacional e tipos diferentes de simulação, tais como elementos finitos, tendo em conta que os resultados destes possam ser convertidos em dados de treinamento similares aos utilizados neste trabalho. Desta maneira, o mesmo tipo de arquitetura pode ser utilizada em diversas aplicações diferentes.

Este trabalho está dividido da seguinte maneira: No Capítulo 2, foi dada uma fundamentação teórica quanto ao dispensador de insumos, assim como os mecanismos físicos que se pretende representar via simulação. No Capítulo 3 foi realizada uma análise bibliográfica, focada tanto em conceitos mais gerais de redes neurais e redes neurais convolucionais, quanto em aplicações onde o mesmo tipo de modelo de substituição foi gerado utilizando-se de aprendizado supervisionado. No Capítulo 4 foram detalhadas as propostas de geração do banco de dados, arquitetura de rede neural, características gerais do modelo de CFD e a análise de erro por variável de treinamento. No Capítulo 5 serão apresentados os resultados obtidos no processo de otimização da rede, análise de erro e análise qualitativa e, por fim, no Capítulo 6 foi abordada a conclusão geral dos resultados obtidos, oportunidades de melhoria e próximos estudos.

2. FUNDAMENTAÇÃO TEÓRICA

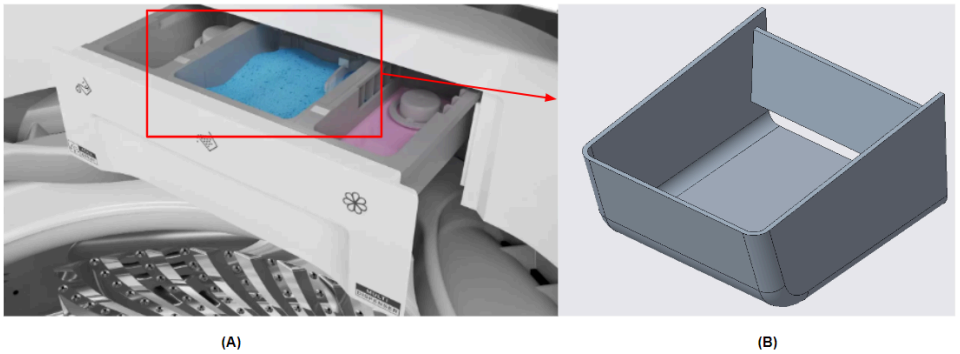
Este capítulo tem como objetivo principal detalhar o mecanismo de funcionamento de um dispensador de gaveta. A compreensão de seu funcionamento é essencial para o entendimento das variáveis selecionadas para o estudo.

O dispensador de insumos é um dos principais sistemas em lavadoras de roupa convencionais. Uma imagem de um dispensador típico utilizado em uma lavadora da marca Brastemp do tipo gaveta pode ser vista na Figura 1(a).

Suas principais funções são, primeiramente, permitir que o consumidor abasteça sua lavadora com a quantidade, tipo e marca de insumos desejados. Além disso, o sistema deve ser capaz de transferir os insumos recebidos para as roupas, considerando-se o momento correto do ciclo. Este processo é realizado através de um escoamento coordenado de água, controlado por válvulas acionadas eletricamente, passando através das câmaras do dispensador.

A Figura 1(b) ilustra uma parte do sistema de dispensamento que será objeto deste estudo. Trata-se do compartimento de sabão em pó, onde o sabão em pó abastecido pelo consumidor é armazenado até ser transferido para a unidade de lavagem.

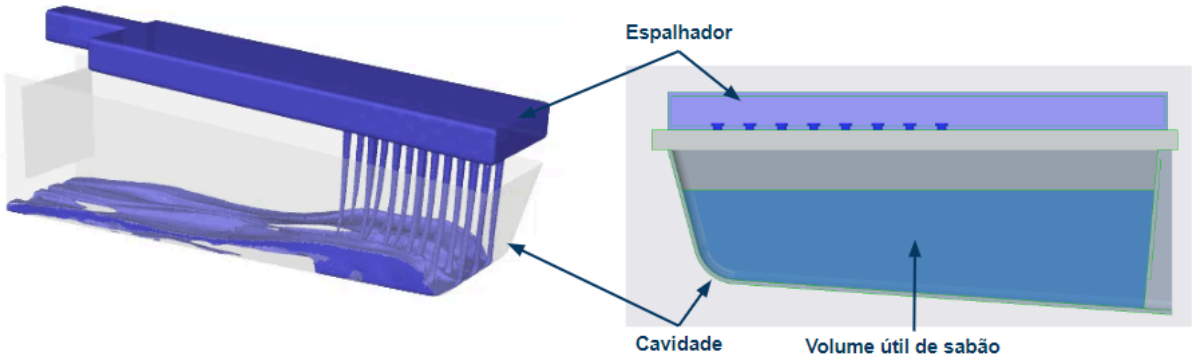
Figura 1 - Dispensador de gaveta comercial.
Em (a), uma imagem do compartimento de sabão em uma lavadora de roupas comercial. Em (b), uma representação simplificada do compartimento isolado.



Fonte: Do autor

O modelo que se pretende representar consiste basicamente em dois componentes principais, denominados aqui de espalhador e cavidade. Estas geometrias estão representadas na Figura 2.

Figura 2 - Geometrias de espalhador e cavidade.



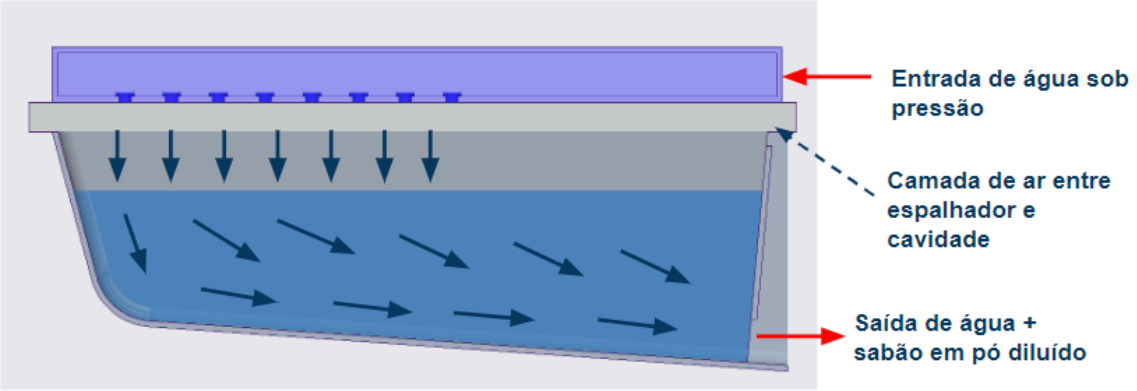
Fonte: Do autor

Em termos funcionais, a função do espalhador é distribuir determinado fluxo de água que entra sob pressão sobre a cavidade, cuja função é realizar o armazenamento e a migração do sabão em pó para o compartimento de lavagem.

O espalhador é um compartimento usualmente selado dotado de um conjunto de orifícios por onde a água sob pressão escoa. Para seu correto funcionamento, o padrão de saídas do espalhador deve garantir um escoamento homogêneo sobre a cavidade, evitando-se regiões secas, onde não há incidência de jatos de água, ou regiões de recirculação, onde as correntes de fluxo de água geram turbilhonamento e não direcionam o insumo para a saída do sistema.

Já a cavidade tem as funções principais de possuir volume mínimo para alojamento de insumo, propiciar uma área adequada para adição de insumo para o consumidor e realizar a migração do insumo minimizando-se resíduos. Em função disso, a cavidade deve possuir uma área de saída suficiente para sustentar o escoamento de água sem transbordar e deve ser capaz de arrastar as partículas de sabão em condições estressadas de uso, funcionando sob baixa pressão de água ou em condições de instalações de uso não ideais, onde o produto pode estar inclinado até 2º no sentido não favorável do dispensamento. Usualmente, o fundo da cavidade possui uma inclinação, exatamente para este fim. A Figura 3 ilustra o sentido de escoamento da água durante o dispensamento de sabão em pó.

Figura 3 - Fluxo de água durante o dispensamento



Fonte: Do autor

O processo de dispensamento de sabão em pó consiste basicamente de dois mecanismos.

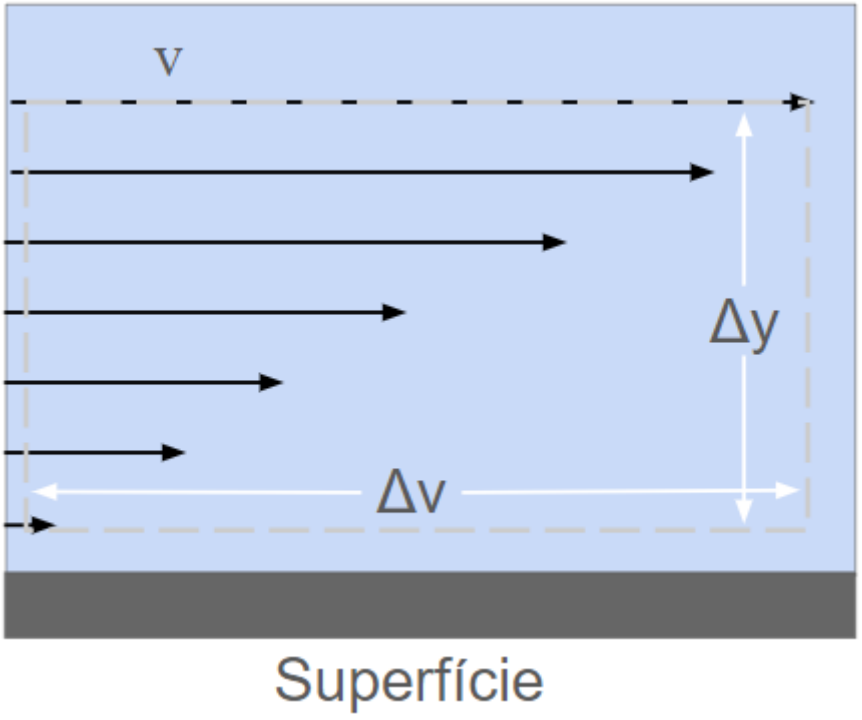
Inicialmente, a água transferida do espalhador para a cavidade deve cobrir o máximo possível a superfície do fundo da cavidade com água. Isto garante que todas as partículas de sabão em pó estejam em contato direto com a água. Como a água é um solvente natural do detergente em pó, este se solubiliza na água, que está sendo transferida para a unidade de lavagem.

Contudo, nem todo o sabão se solubiliza na água, em função da composição do sabão em pó conter compostos, tais como partículas abrasivas que são importantes para a remoção de manchas e sujidade das roupas. Desta maneira, a migração destas partículas para a unidade de lavagem se dá pelo arraste das partículas sobre a superfície da cavidade em direção ao fundo do compartimento. A força de cisalhamento que age sobre determinado fluido pode ser expressada, considerando-se simplificações para um sistema de duas placas planas paralelas, pela equação da lei de Newton para viscosidade, expressa na Equação 1(MUNSON, 1995) :

$$\tau = \mu \frac{\Delta v}{\Delta y}, \qquad (1)$$

onde τ representa a tensão de cisalhamento agindo sobre a superfície, μ representa uma propriedade do fluido denominada viscosidade dinâmica e o termo $\frac{\Delta v}{\Delta y}$ representa a variação de velocidade de um fluido em relação à superfície, até o ponto em que a velocidade do fluido se torna constante. A Figura 4 abaixo representa uma vista esquemática destas variáveis.

Figura 4 - Diagrama das grandezas físicas que geram cisalhamento sobre uma superfície.



Fonte: Do autor

A viscosidade μ é uma propriedade intrínseca do fluido que está interagindo com a superfície. A água, no caso, tem uma viscosidade dinâmica específica, que é afetada principalmente pela temperatura. A viscosidade dinâmica da água a 7°C, por exemplo, é de 1.42 mPa.s, enquanto a 37° é de 0.7 Pas (MUNSON e ROY, 2013).

Desta maneira, a viscosidade é um ruído do ponto de vista do desenvolvimento do componente, cabendo ao engenheiro maximizar a velocidade do escoamento sobre aquela superfície, visando aumentar o cisalhamento sobre aquela superfície.

Considerando-se que a viscosidade do fluido é de certo modo incontrolável, já que o produto é comercializado em regiões em que a temperatura e composição

da água no geral variam, a margem de manobra do engenheiro é desenvolver o componente de modo a maximizar a área de cobertura da água sobre a superfície onde o sabão em pó é alojado e maximizar a velocidade da água uniformemente sobre esta superfície, de modo a ser capaz de arrastar as partículas depositadas em qualquer região da superfície.

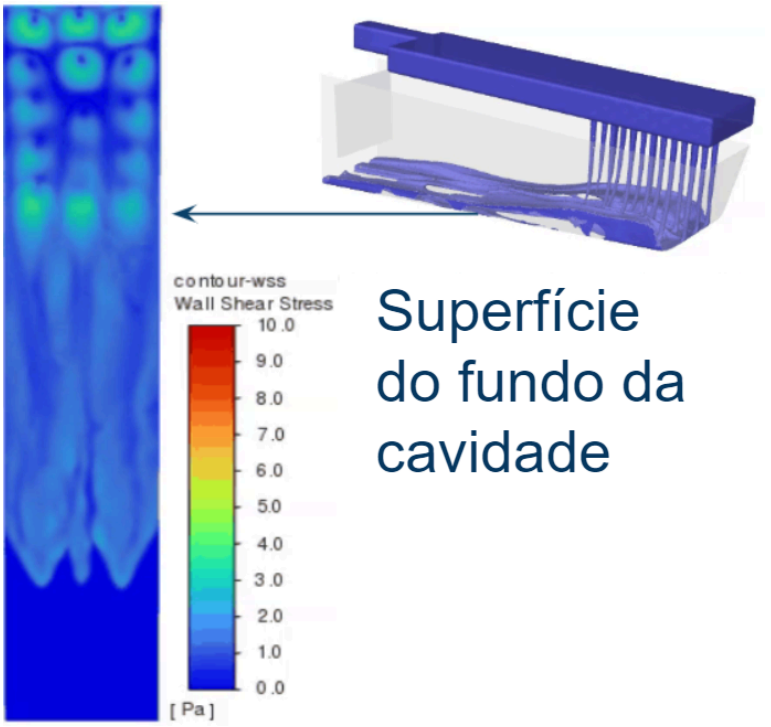
Para o estudo em questão, duas variáveis foram consideradas. A primeira, denominada fração volumétrica ou VOF, do inglês *volumetric fraction*, diz respeito a qual percentual de determinado volume de controle contém água em relação às partículas de ar. Para a simulação de CFD, que será detalhada na Seção 3.2, valores de VOF acima de 50% indicam que determinado volume de controle contém água. Desta maneira, mede-se a área molhada da superfície do fundo da cavidade, o que significa que a água está cobrindo todo o fundo da cavidade de dispensamento.

A segunda variável que foi medida é a tensão de cisalhamento τ, que conforme mencionado anteriormente está relacionada à força cortante de arraste das partículas que age sobre a superfície.

Estas duas características são extraídas da superfície do fundo da cavidade, sob a qual as partículas de sabão ficam alojadas. Entende-se que o mapeamento contínuo destas duas grandezas sob esta superfície deve ser suficiente para ter uma boa representatividade da ocorrência de resíduo de sabão no sistema, uma vez que esta sempre ocorrerá sobre esta superfície.

A Figura 5 abaixo demonstra a superfície da qual os campos de cisalhamento e VOF serão extraídos para cada geometria de dispensador.

Figura 5 - Extração de perfil escalar de superfície do fundo da cavidade



Fonte: Do autor

Como última consideração, vale destacar que para o domínio em questão, apenas os meios água e ar foram estudados, excluindo-se a presença do sabão. Entende-se que a aproximação é razoável, pois, em regime permanente, apenas uma quantidade residual de sabão deveria permanecer na superfície, o que não deveria influir negativamente no comportamento físico do escoamento.

Além do mais, a adição de um meio sólido, como sabão em pó, à simulação, traz uma complexidade alta à sua resolução, uma vez que o modelamento da interação de particulado sólido com um meio líquido não é trivial, o que ainda é exacerbado por ocorrer solubilização do sabão na água durante o escoamento.

Em iniciativas a serem realizadas nos próximos anos, pretende-se realizar medições diretas do acúmulo de partículas de sabão em pó sobre superfícies de dispensadores atualmente em comercialização em lavadoras correntes. Este estudo terá como objetivo a medição direta do volume e localização residual do sabão em pó sobre esta superfície. Tendo esta informação, será possível avaliar se os campos escalares de cisalhamento e fração volumétrica representam de maneira eficiente a ocorrência de resíduo em cada elemento do domínio.

3. ANÁLISE BIBLIOGRÁFICA

Neste capítulo se encontra uma revisão relacionada aos temas pertinentes a este trabalho e que contribuem para a criação do modelo proposto. A pesquisa foi realizada na plataforma *Science Direct* (HUNTER,1998), utilizando as palavras-chave *deep learning*, convolutional neural networks, computational fluid dynamics, *multi-objective optimization* e *adversarial generative networks* .

Na Seção 3.1 é apresentada uma breve introdução à fluidodinâmica computacional, incluindo o método de discretização, as equações regentes do escoamento e as limitações que ainda existem quanto ao uso do CFD para modelos de otimização.

As Seções 3.2 e 3.3 são dedicadas a detalhar, respectivamente, o esquema de funcionamento dos dois tipos de redes neurais mais aplicadas dentro do escopo do trabalho, as redes neurais perceptron de multicamadas (*multilayer perceptron)* e as redes neurais convolucionais.

Posteriormente, na Seção 3.4 são revisados trabalhos relacionados ao uso de redes neurais como modelos de substituição que utilizam as respostas dos exemplos executados no software de CFD como dados de treinamento, estimando a variável resposta. Também serão revisados trabalhos em que além de utilizar apenas a variável de resposta da variável de performance, os próprios campos escalares, resultados das simulações de CFD, foram utilizados como dados de treinamento, visando a geração de modelos que, além de estimar a variável de desempenho, possuem como saída os próprios campos escalares e vetoriais, o que significa intrinsecamente aprender propriedades relacionadas à interação fluido-estrutura.

Estas arquiteturas, de fato, têm a capacidade de substituir a simulação de CFD e aprender propriedades intrínsecas ao escoamento, como regiões de recirculação, turbulência, regimes transientes, etc (MALALASEKERA et al., 1995).

Na Seção 3.5 será realizado um resumo dos temas abordados no Capítulo 2.

3.1. Introdução à fluidodinâmica computacional

Um dos campos da engenharia em que mais há potencial para aplicação de técnicas de aprendizado de máquina, ou *deep learning*, é a área de Fluidodinâmica Computacional, ou CFD (*computational fluid dynamics*).

A fluidodinâmica computacional, juntamente com a análise de elementos finitos (FEA) (ALVES FILHO, 2018), são conhecidos como métodos de simulação numérica (MALALASEKERA, 1995), que são esquemas numéricos dedicados a resolver equações algébricas, sistemas de equações e sistemas de equações diferenciais, de maneira a simplificar um domínio complexo, regido por fenômenos cujas equações diferenciais são insolúveis analiticamente, em equações algébricas, que podem ser resolvidas numericamente.

O escoamento de fluidos, sejam eles gases ou líquidos, sobre um meio sólido ou em relação a outro meio fluido, são regidos por equações diferenciais complexas, relacionadas à conservação de massa, quantidade de movimento e conservação de energia (MUNSON e ROY, 2013).

Os fenômenos que são descritos por tais equações são diversos, tais como reações químicas, combustão, escoamento de fluidos sobre estruturas (aerofólios, cascos de navio, turbinas). Este tipo de escoamento é regido por um conjunto de equações denominadas Navier-Stokes (MUNSON e ROY, 2013), sendo elas a equação de continuidade (Equação 2), as equações de conservação de momento considerando as três direções cartesianas (Equações 3 a 5), e a equação de balanço de energia (Equação 6). As equações abaixo consideram calor específico constante (BIRD et al., 1960):

$$\frac{\partial \rho}{\partial t} + \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} = 0 \tag{2}$$

$$\frac{\partial v_x}{\partial t} + \frac{\partial \rho v_x v_x}{\partial x} + \frac{\partial \rho v_y v_x}{\partial y} + \frac{\partial \rho v_z v_x}{\partial z} = \rho g_x - \frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left(\mu \frac{\partial v_x}{\partial x}\right) + \frac{\partial}{\partial x}\left(\mu \frac{\partial v_x}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu \frac{\partial v_x}{\partial z}\right) \tag{3}$$

$$\frac{\partial v_y}{\partial t} + \frac{\partial \rho v_x v_y}{\partial x} + \frac{\partial \rho v_y v_y}{\partial y} + \frac{\partial \rho v_z v_y}{\partial z} = \rho g_y - \frac{\partial p}{\partial y} + \frac{\partial}{\partial x}\left(\mu \frac{\partial v_y}{\partial x}\right) + \frac{\partial}{\partial x}\left(\mu \frac{\partial v_y}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu \frac{\partial v_y}{\partial z}\right) \tag{4}$$

$$\frac{\partial v_z}{\partial t} + \frac{\partial \rho v_x v_z}{\partial x} + \frac{\partial \rho v_y v_z}{\partial y} + \frac{\partial \rho v_z v_z}{\partial z} = \rho g_z - \frac{\partial p}{\partial z} + \frac{\partial}{\partial x}\left(\mu \frac{\partial v_z}{\partial x}\right) + \frac{\partial}{yx}\left(\mu \frac{\partial v_z}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu \frac{\partial v_z}{\partial z}\right) \tag{5}$$

$$\frac{\partial \rho C_p}{\partial t} + \frac{\partial \rho v_x C_p T}{\partial x} + \frac{\partial \rho v_y C_p T}{\partial y} + \frac{\partial \rho v_z C_p T}{\partial z} = K\left(\frac{\partial T}{\partial x^2} + \frac{\partial T}{\partial y^2} + \frac{\partial T}{\partial z^2}\right) + \ldots$$

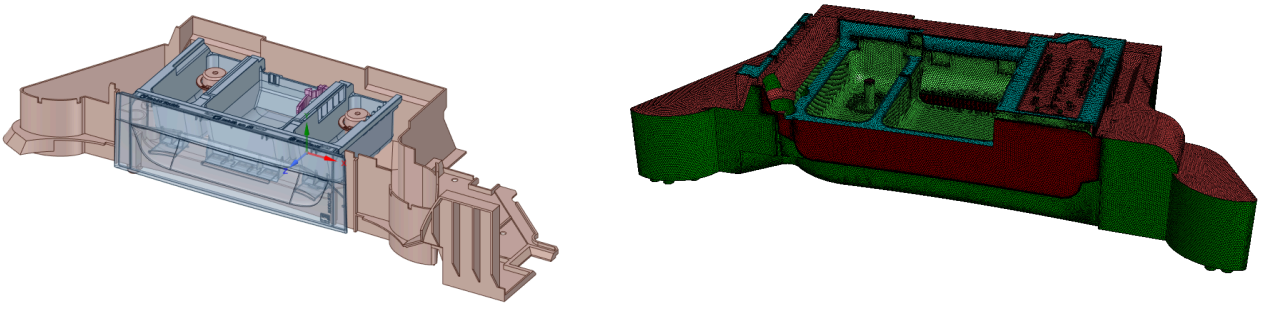
$$\ldots 2\mu\left\{\left(\frac{\partial v_x}{\partial x}\right)^2 + \left(\frac{\partial v_y}{\partial y}\right)^2 + \left(\frac{\partial v_z}{\partial z}\right)^2\right\} + \mu\left\{\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right)^2 + \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x}\right)^2 + \left(\frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y}\right)^2\right\}, \tag{6}$$

em que *p* representa pressão, *t* é o tempo, *T* representa temperatura, *v* é velocidade, *g* representa gravidade, ρ é a densidade do fluido, μ é a sua viscosidade dinâmica, C_p é seu calor específico e *K* é a condutividade térmica. Os subscritos em cada termo representam a direção cartesiana considerada, enquanto ∂ representa a derivada parcial. Os termos entre chaves estão relacionados à dissipação viscosa do fluido. Em grande parte das aplicações, com exceção de situações específicas, tais como o escoamento laminar entre duas placas, estas equações não possuem solução analítica direta, sendo necessário métodos numéricos e simplificações, em dinâmica dos fluidos denominados de modelos de turbulência, para sua resolução (MUNSON e ROY, 2013).

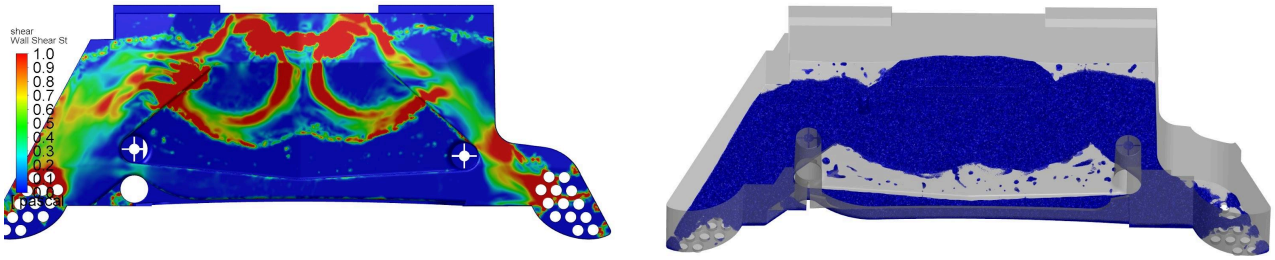
As equações diferenciais mencionadas acima, em geral, não possuem solução analítica, sendo que sua resolução com o advento da fluidodinâmica computacional se deu pelos denominados métodos de discretização (MALALASEKERA, 1995). A discretização consiste em substituir-se um determinado domínio contínuo por um conjunto de elementos mais simples e discretos ao invés de contínuos. Desta maneira, o espaço contínuo original é substituído por um conjunto de elementos mais simples. Entre os métodos de discretização mais aplicados existe o método de elementos finitos e o método de volumes finitos, este último amplamente utilizado em simulações de fluidodinâmica. A discretização de um domínio é ilustrada abaixo na Figura 6.

Figura 6 - Exemplo de domínio tridimensional discretizado por malha e resultados obtidos em regime permanente.

Em (a), uma vista isométrica do dispensador de gaveta. Em (b), uma vista da malha gerada a partir deste domínio. Em (c), o perfil de cisalhamento em uma das superfícies do domínio e em (d) o perfil de fração volumétrica da superfície do fundo do dispensador.



(a) (b)



(c) (d)

Fonte: Figura do autor

Na Figura 6 é possível verificar um dispensador químico do tipo gaveta em sua geometria 3D e sua representação para simulação de CFD. Na Figura 6(a), o modelo CAD (*Computer Aided Design*) é utilizado de entrada. Na Figura 6(b), o modelo é discretizado em elementos, em um processo conhecido como geração de malha (MALALASEKERA, 1995). Os elementos de malha devem ter tamanho suficiente para representar corretamente o comportamento do fluido. Este modelo em questão, que fisicamente possui dimensões aproximadas de 60 cm x 15 cm, possui aproximadamente 2.000.000 elementos e possui tempo de simulação de aproximadamente 12 horas em um computador de alto desempenho. Na Figura 6(c) é mostrado o perfil de tensão de cisalhamento na superfície da peça em regime permanente, enquanto na Figura 6(d) é mostrada a área molhada em regime permanente. Estas são duas variáveis que são analisadas como resposta da simulação.

O objetivo final de uma simulação fluidodinâmica é obter o entendimento sobre a interação entre a geometria que se está estudando e o fluido, de maneira a se obter o desempenho hidrodinâmico desejado, seja ele relacionado à otimização ou minimização de determinada variável resposta.

A principal vantagem dos métodos de simulação computacional é a substituição da necessidade de se prototipar cada conceito a ser estudado em um determinado desenvolvimento por um modelo computacional equivalente, uma vez que o estudo de correlação demonstra aderência entre as medidas experimentais e o modelo computacional.

Em função desta capabilidade, os modelos de CFD permitem otimizações proativas do *design*, de maneira a reduzir as interações de desenvolvimento e melhorar o desempenho e previsibilidade da etapa de desenvolvimento. Entre as vantagens da aplicação de técnicas de otimização geométrica, destacam-se (HARRIES, 2020): melhor entendimento dos limites funcionais das variáveis independentes, criação de produtos com desempenho superior, minimização de riscos durante a etapa de desenvolvimento, redução dos custos de projeto e redução do *time-to-market.*

Entre os principais tipos de otimização geométrica, destacam-se a otimização completamente parametrizada, a otimização semi-parametrizada e a otimização não parametrizada (HARRIES, 2020).

Ainda assim, devido à complexidade da física regente do escoamento de fluidos, simulações de CFD podem levar dias até a convergência, o que torna a testagem de centenas de combinações de variáveis inviáveis.

Desta maneira, o gargalo da etapa de desenvolvimento envolvendo CFD é que mesmo modelos numéricos de complexidade média necessitam de centenas de simulações custosas computacionalmente, que levam dias para serem concluídas (XU et al., 2021).

Além disso, por diversas vezes, o domínio que se deseja otimizar é amplamente conhecido na aplicação e a mesma simulação é executada para cada novo projeto. Neste âmbito, as aplicações de aprendizado profundo relacionadas a simulações de CFD tem focado na utilização de arquiteturas baseadas em aprendizado supervisionado que, utilizando-se de uma modelos iniciais previamente simulados em um software de CFD, estima a resposta de uma variável de performance ou, em aplicações mais recentes, no aprendizado de fato dos campos de pressão, velocidade, etc. Estes modelos são comumente denominados de modelos de substituição, ou *surrogate models* (PEHERSTORFER et al., 2018*;* KOZIEL e LEIFSSON, 2013).

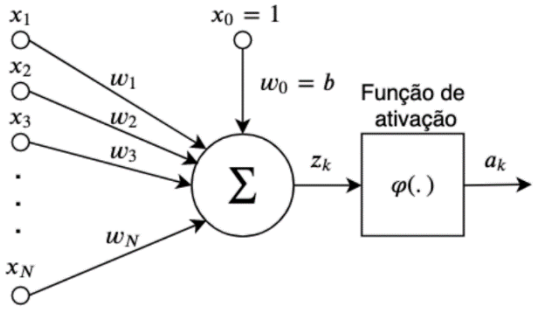
Os primeiros tipos de modelos de substituição tratavam a simulação de CFD como uma caixa preta e construiam diretamente uma correlação entre uma variável de performance que se deseja medir e a variável resposta, sem se preocupar com as não linearidades físicas envolvidas no processo. Métodos tradicionais de modelos de substituição são os métodos de krigagem (*Kriging*) (KRIGE, 1951) e mapeamento de variedades (*manifold mapping*) (ECHEVERRÍA et al., 2008). Em substituição aos

modelos tradicionais, a aplicação de redes neurais profundas (DNN - *deep neural networks*), especialmente MLPs, surgiram como um tipo importante de modelo, por captar bem as não linearidades do sistema (BOUHLEL et al., 2020; LIU et al., 2019).

3.2. Perceptron multicamadas

O conceito de neurônio artificial como aproximação de um neurônio biológico foi proposto em McCulloch e Pitts(1943). A Figura 7 ilustra a estrutura do neurônio perceptron.

Figura 7 - Modelo Perceptron



Fonte: Mcculloch e Pitts(1943)

Na figura 7 acima, *X1* a *Xn* representam os neurônios de entrada da rede. *W1* a *Wn* representam os pesos que relacionam os neurônios a um determinado neurônio da camada seguinte. *X0* e *W0* são denominados neurônios de viés e seu respectivo peso. A variável *φ* representa a função de ativação, que recebe as entradas relacionadas a somatória dos neurônios multiplicados por seus pesos, representado pela variável *zk* e aplica uma determinada função de transformação para gerar a saída *Ak*. A função de ativação pode ser entendida como um operador que gera uma resposta *Y* a partir da somatória das variáveis de entrada *X*.

A organização de vários neurônios perceptron em camadas distintas ficou conhecida na literatura como a arquitetura de redes Multilayer Perceptron (RUMELHART, 1986). O uso de redes de neurônios com mais de uma camada visava resolver os problemas encontrados na classificação de dados não linearmente separáveis como a emulação de uma porta ou-exclusiva (XOR). Redes com essa arquitetura aliada ao treinamento com o algoritmo *backpropagation* são capazes de classificar corretamente dados não linearmente separáveis

(GOODFELLOW et al., 2014). A equação das ativações e pesos ponderados de uma rede MLP é representada abaixo nas Equações 7 e 8 (ROSENBLATT, 1961):

$$z_j^l = \sum_{k=1}^{N} (\omega_{jk}^l * a_k^{l-1}) + b_j^l , \quad (7)$$

$$a_j^l = \varphi(z_j^l), \quad (8)$$

em que N é o número de entradas do neurônio, ω_{jk}^l representa o peso vindo do neurônio *k* da camada *(L − 1)* para o neurônio *j* da camada *L*, e a_{lk}, b_j *e* z_j correspondem respectivamente a ativação, viés e peso ponderado do neurônio *k* da camada *L*.

De maneira geral, as redes neurais MLP possuem amplo histórico de aplicação em diversos setores, especialmente devido à sua alta capacidade de representação de funções de alta complexidade. Redes deste tipo com uma camada completamente conectada entre a camada de entrada e a camada de saída, usualmente denominada de camada oculta, tem a capacidade de representar qualquer tipo de função algébrica. A arquitetura pode ser utilizada para problemas de classificação ou regressão, a depender das funções de ativação utilizadas.

A estimativa da variável resposta a partir das variáveis de entrada, propagada pela rede através das camadas ocultas e função de ativação é denominada de *forward propagation.* Uma vez que a variável resposta de determinada rede depende das variáveis de entrada (independentes), dos pesos e das funções de ativação, o que permite a rede ser capaz de aproximar funções com a atualização dos pesos, a partir da estimativa do erro obtido entre o resultado estimado pela rede durante a etapa de *forward propagation* e os resultados reais para determinado treinamento.

Após o cálculo dos erros relacionados às estimativas, algum algoritmo de treinamento é utilizado para atualizar os pesos em relação aos erros calculados, retro propagado da camada de saída para a camada de entrada. O algoritmo mais usual empregado na aplicação é o algoritmo de descida do gradiente estocástico (SGD - *stochastic gradient descent*) , onde uma taxa de aprendizado fixa atualiza o valor dos pesos a cada época, de modo a cada época minimizar o erro entre as estimativas do modelo e o valor real. A minimização das estimativas ocorre

atualizando-se, época a época, os pesos que conectam os neurônios da camada de saída até a camada de entrada (CAUCHY, 1847).

3.3. Redes neurais convolucionais

Redes neurais convolucionais são tipos de redes neurais que foram desenvolvidas para processamento de dados de treinamento que possui uma forma de entrada de dados em forma de grade. Exemplos incluem dados como séries temporais, imagens, vídeos ou volumes 3D. As redes obtiveram tremendo sucesso em diversas aplicações práticas (LECUN, 1989).

O grande diferencial deste tipo de arquitetura é que, através de uso de camadas de convolução e *pooling*, que serão detalhadas posteriormente neste capítulo, a rede necessita de apenas poucas camadas completamente conectadas entre a camada de entrada e a camada de saída para o treinamento. Ao invés disso, através da aplicação de filtros de convolução, ela sintetiza o conteúdo semântico dos dados de entrada através da convolução de filtros sobre os dados de entrada e, através do processo de *polling,* torna o conteúdo semântico invariável a pequenas variações daquele conteúdo semântico. Isto torna o processo de treinamento mais eficiente e rápido (GOODFELLOW et al., 2016).

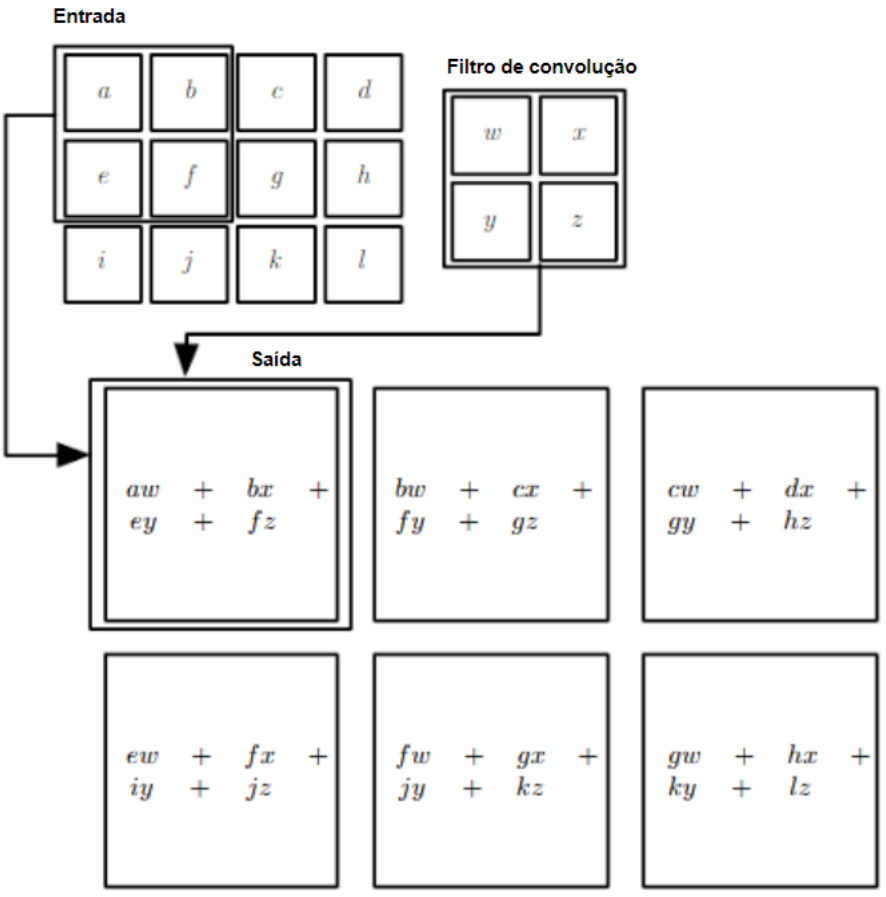
Considera-se um caso, por exemplo, onde se pretende classificar se determinada classe de imagem digital contém um atributo de determinada classe. Considerando-se uma imagem por exemplo que tenha tamanho 800x600 pixels, cada pixel seria tratado como um neurônio de entrada em uma MLP tradicional, resultando em uma camada de entrada de 480.000 neurônios. Cada camada oculta precisaria ter um tamanho na casa de centenas milhares de neurônios para carregar o conteúdo semântico, o que torna o processo de treinamento lento.

Uma CNN, neste mesmo cenário, aplicaria uma série sucessiva de operações de convolução e pooling na imagem de entrada 800 x 600, de modo que a cada camada, uma matriz resultante de dimensão usualmente menor que o dado de entrada carregaria o significado semântico necessário da camada anterior, até que o tamanho do tensor após esta sucessão de camadas seja treinável com uma MLP convencional. Usualmente, redes convolucionais de classificação possuem adjacentes à camada de saída, camadas completamente conectadas.

Além do mais, em uma tarefa de classificação de imagens, nem todos os pixels possuem significado semântico, o que agrega processamento computacional desnecessário para o objetivo.

A convolução é ilustrada na Figura 8 abaixo. O processo de convolução consiste em utilizar um filtro, com tamanho usualmente menor que os dados de entrada. O filtro, comumente denominado de *kernel*, é utilizado para detectar padrões dentro de um grupo reduzido de pixels, aplicando uma operação de multiplicação elemento por elemento.

Figura 8 - Operação de convolução



Fonte: Goodfellow (2016)

Durante o processamento de uma imagem, por exemplo, os dados de entrada podem ter milhões de pixels, mas é possível detectar características críticas, como bordas, com *kernels* que ocupam apenas uma dezena ou centena de pixels (GOODFELLOW, 2016). Além disso, o processamento da saída é muito mais eficiente.

Diferentemente de uma rede neural MLP, onde os pesos aprendidos são os pesos das ligações completamente conectadas entre os neurônios, em uma rede convolucional o próprio *kernel* de convolução é atualizado pelo processo de treinamento, tornando-se capaz de detectar padrões. Usualmente, o processo de convolução se dá iterando o filtro de convolução sobre toda a imagem que se pretende realizar o processo de convolução.

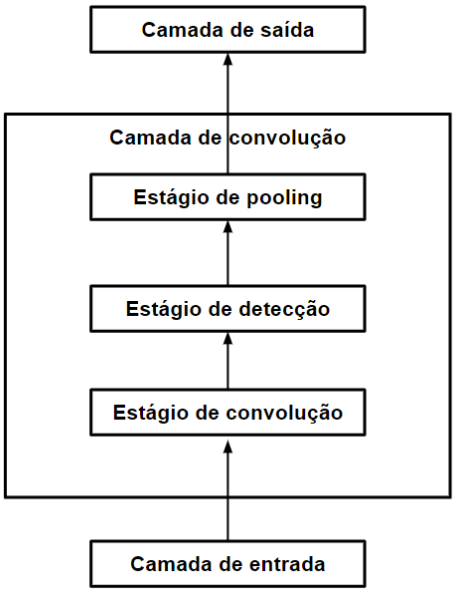
Além da camada de convolução, outros elementos das redes convolucionais são as camadas de *pooling*. Uma camada de *pooling* substitui determinada saída de uma camada por um resumo estatístico das variáveis de entrada. Usualmente, em redes convolucionais, utilizam-se as funções *Max Pooling*, que mantém o maior valor dentro do kernel, e *average pooling*, que faz uma média aritmética dos valores do *kernel*.

Uma função da camada de *pooling* é tornar o meio invariante a pequenas translações, o que significa que dois conjuntos com pequenos deslocamentos ou rotações de pixels entre eles tendem a ter o mesmo resultado após a aplicação da camada. Outra função da camada de *pooling* se dá justamente pela sua capacidade de sumarizar informação de um determinado conjunto de elementos (GOODFELLOW, 2016).

Isto se dá realizando-se a operação de *pooling* a cada K pixels ou elementos ao invés de iterando posição a posição. Desta maneira, um determinado conjunto de dados de entrada MxN após um processo de *pooling* a cada K neurônios apresentará dimensões finais (M/K)x(N/K). Desta maneira, a camada de *pooling* permite a redução da dimensionalidade dos dados de entrada com pouca perda de informação.

De maneira geral, a camada de convolução pode ser representada de acordo com a Figura 9.

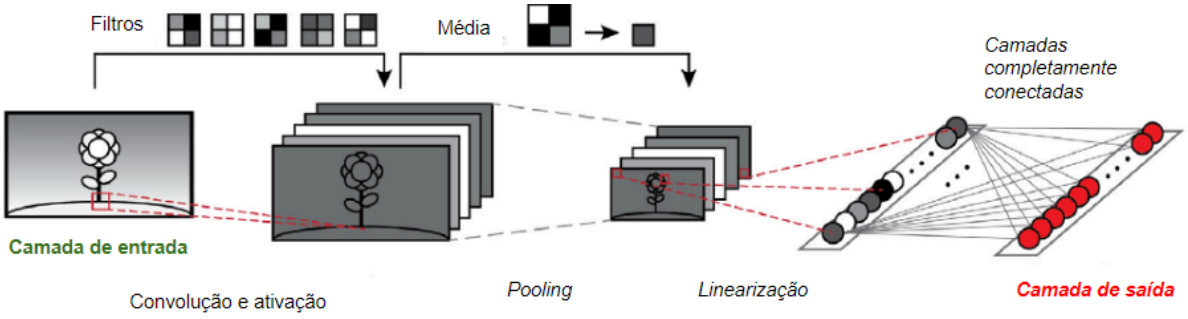
Figura 9 - Diagrama de camada convolucional



Fonte: Goodfellow, 2016

No estágio de convolução, um conjunto de filtros é iterado sobre os elementos de entrada. O tensor de saída desta operação passa por uma função de ativação, elemento a elemento, de modo a gerar um tensor de tamanho igual à entrada, mas com o objetivo de ressaltar as não linearidades relativas à entrada após convolução. Este tensor de saída é usualmente denominado mapa de características. Posteriormente, dentro da camada, o mapa de características passa pelo estágio de *pooling*, reduzindo a dimensão dos dados de entrada. A Figura 10 ilustra uma estrutura geral de rede neural convolucional.

Figura 10 - Rede neural convolucional



Fonte: Lalonde et al. (2021).

As arquiteturas de rede convolucional aplicam usualmente um conjunto de camadas de convolução e *pooling* sucessivas, principalmente com o objetivo de

reduzir o tamanho dos dados de entrada. Após um determinado número de camadas de convolução, a menor dimensão dos dados de entrada permite o processamento por uma rede MLP convencional. Desta maneira, usualmente as redes neurais convolucionais possuem em suas últimas camadas redes neurais MLP completamente conectadas. Os dados passam por um processo denominado linearização, onde basicamente cada conjunto de entrada de tamanho MxN se torna um conjunto do tipo (MxN)x1, de maneira a ser processada pela rede MLP.

Assim como em uma rede MLP convencional, o tipo de tarefa a ser realizada pela rede depende das funções de ativação atribuídas à arquitetura do modelo, tanto das camadas completamente conectadas quanto das camadas de convolução.

Dois parâmetros importantes da etapa de convolução são os denominados *padding* (preenchimento) e *stride* (passo).

O *padding* é um processo essencial em redes convolucionais que possuem diversas camadas de convolução em série. Em um processo de convolução sem padding, a imagem convolucional possui menos pixels do que a imagem de entrada, uma vez que o *kernel* percorre a imagem posição a posição até cobrir todos os pixels da imagem original. Na Figura 8, por exemplo, a imagem do formato 4x3 passa a ter forma 3x2 após a convolução. A perda de pixels por convolução depende do tamanho do filtro que se está utilizando. O processo de *padding* consiste em adicionar pixels nas bordas dos dados de entrada antes da convolução de maneira que, após o processo de convolução, o tamanho do tensor de saída seja preservado. Em redes que possuem diversas camadas de convolução, o *padding* impede que o tensor de saída se torne excessivamente pequeno.

Já o processo de *stride* denota o passo que o filtro de convolução será aplicado sobre a imagem original. Em uma convolução com *stride* unitário, a cada movimento do filtro ele desloca uma posição na imagem original até a borda. Caso o *stride* seja maior do que um, o filtro de convolução avança mais de uma posição em cada interação da convolução. Uma convolução sobre uma imagem NxN com *stride* = 2 terá tamanho (N/2)x(N/2) após a convolução, e assim sucessivamente. Quanto menor o *stride* utilizado, mais rico o conteúdo que a rede carrega para a próxima camada, mas maior o custo computacional.

3.4. Redes neurais para previsão fluidodinâmica

Em Lalonde et al. (2021), os autores realizaram um estudo de diversas arquiteturas baseadas em redes neurais relacionadas a modelos de substituição (*surrogate models*, do inglês). Os cálculos aerodinâmicos de uma turbina eólica consistem em determinar as cargas nas pás a partir do vento que são afetadas pela posição do aerofólio, geometria e rotação. Há diversos métodos numéricos para realizar tais estimativas. Porém, eles geralmente geram uma resposta rápida com baixa acuracidade ou uma resposta mais precisa com tempo e custo computacional elevados, principalmente com o uso de CFD. Os autores analisaram três tipos principais de arquitetura para realizar a otimização, um MLP convencional, uma rede LSTM-*Long Short-Term Memory* (GRAVES, 2012) e uma rede neural convolucional (CNN).

Quanto ao treinamento aplicado à capacidade de previsão da rede, os autores destacam que uma rede bem treinada com um número suficientemente grande de dados tem capacidade de prever com precisão adequada a performance de determinada geometria, porém tem sua performance restrita aos dados de treinamento utilizados. Para a aplicação em questão, como os autores possuíam interesse no mapeamento quantitativo e contínuo de diversas variáveis, as redes usaram função erro médio quadrático normalizado.

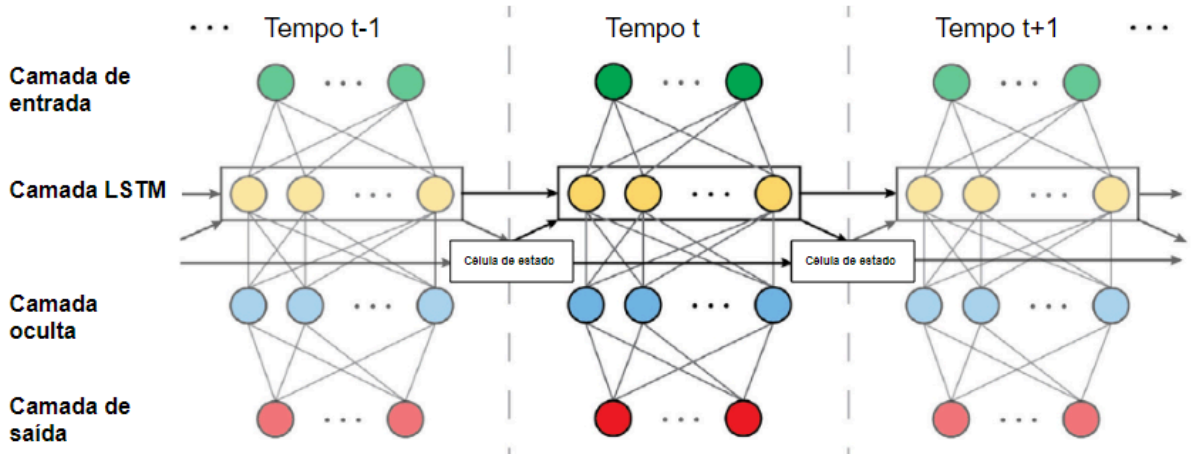
A rede LSTM é uma forma de rede neural recorrente que é especializada em processar dados ao longo do tempo. Ao invés de considerar cada intervalo de tempo como um ponto individual, característica típica de uma MLP convencional, ela também utiliza variáveis resposta de tempos anteriores para a estimativa da variável resposta (GRAVES, 2012). As LSTMs são um tipo de rede recorrente desenvolvida para ter memória a longo prazo. Em função disso, teoricamente deveriam apresentar performance superior em previsões relacionadas a intervalos de tempo.

A diferença em arquitetura de uma rede LSTM para um *multilayer perceptron* é que cada camada da rede LSTM, além das camadas completamente conectadas, possuem uma camada de célula de estado que lembra passos anteriores, que influenciaram na resposta, utilizando funções de ativação similares a uma MLP convencional.

A principal desvantagem da LSTM é que se a relação temporal em dados vizinhos não for significativa, a complexidade e variáveis adicionadas para otimizar o

treinamento podem resultar em uma rede menos precisa do que modelando cada intervalo de tempo como uma variável independente. A rede LSTM é ilustrada na Figura 11.

Figura 11 - Ilustração de rede LSTM



Fonte: Lalonde et al. (2021).

Quanto à aplicação de CNNs, apesar das redes serem tipicamente utilizadas em dados não-estruturados em forma de matriz, principalmente imagens, elas têm capacidade de serem utilizadas em sequências temporais.

Para geração dos dados de treinamento, os autores utilizaram um software de modelamento, considerando uma turbina padrão de 5 Megawatts. Foi gerado um conjunto de variações de velocidade do vento, velocidade do rotor e ângulo da hélice. Eles utilizaram o Software TurbiSim para gerar as séries temporais de velocidade e direção do vento. Foram gerados perfis de vento de 600s, executando 5 repetições em 23 combinações dos fatores mencionados acima.

Os pontos de resposta, considerando as séries temporais, equivalem a um total de 34,5 milhões de pontos individuais adquiridos. Os dados foram obtidos em 18 pontos individuais ao longo do eixo de cada pá da turbina através do Software ElastoDyn. As variáveis resposta do estudo são as cargas aerodinâmicas impostas na turbina em 18 pontos medidos a cada intervalo de tempo do estudo.

Para o treinamento, os dados foram normalizados escalonando-os entre os valores máximo e mínimo obtidos entre 0 e 1. Os dados foram divididos em 60%-20%-20% do total para treino, validação e teste, respectivamente. Os autores executaram uma otimização de hiperparâmetros para cada arquitetura testada.

Para a arquitetura da CNN, os autores consideraram que cada variável de entrada consistia no resultado de determinado tempo, além dos 8 resultados de tempos anteriores. Uma camada de convolução/*pooling* foi utilizada, seguida de 2 camadas completamente conectadas. O treinamento foi realizado por 150 épocas, utilizando o algoritmo de descida do gradiente com taxa de aprendizado de 0,01.

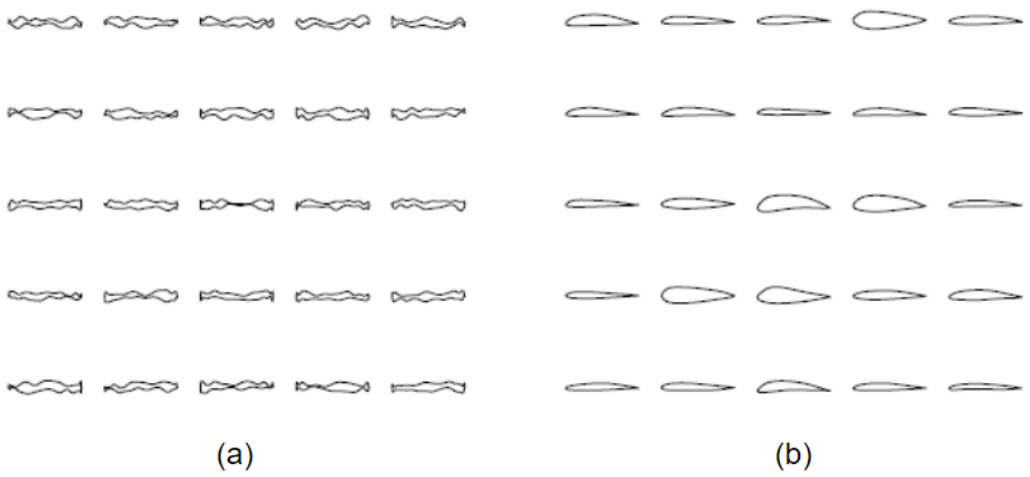
Entre as redes testadas, os autores testaram duas arquiteturas de MLP, que denominaram MLP completa e MLP reduzida, a rede LSTM, uma CNN para dados 1-D e uma rede MLP adaptada com parâmetros de intervalos de tempo anteriores. A diferença da MLP reduzida é que a rede reduzida não recebeu os parâmetros de geometria deformada na previsão. Esta estratégia visava verificar se seria possível obter resultados da MLP completa, mas necessitando de menos esforço computacional na geração dos dados de treinamento. Os melhores resultados foram obtidos pela rede MLP completa e pela rede CNN. Os autores ressaltaram que, apesar da CNN gerar uma quantidade muito maior de dados de entrada (por considerar os 8 intervalos de tempo anteriores), em função das operações de convolução e *pooling* ela conseguiu lidar com os dados, gerando um erro médio quadrático médio de apenas 0,66% em todos os tempos, para todos os pontos. A performance obtida pela rede MLP foi de 1,11%. A rede LSTM obteve um erro médio quadrático normalizado (NRMSE) de 2,45%. As estruturas das camadas de cada rede podem ser acessadas no artigo. A rede MLP reduzida obteve resultados inferiores a MLP completa, apresentando um erro de 5,78%.

Em Du et al. (2021), o objetivo do trabalho foi gerar um modelo de substituição capaz de representar de maneira eficiente os coeficientes de sustentação (Cl) e arrasto (Cd) de um aerofólio para asas de aeronaves. A principal inovação a respeito deste trabalho se deve à geração da população inicial a ser utilizada para treinamento do modelo de substituição. A utilização de equações paramétricas, como a equação de perfis NACA (*National Advisory Committee for Aeronautics*) (ABBOTT et al., 1952) geram restrições no espaço de inferência a ser analisado, enquanto parametrizações mais generalistas (tais como a técnica de B-spline (DE BOOR, 1978) e curvas de Bézier (MORTENSON, 1999) acabam gerando diversos indivíduos irreais, que naturalmente são inúteis do ponto de vista prático e computacional, além de ocupar tempo do próprio algoritmo.

A fim de superar as desvantagens inerentes ao processo de geração aleatória causadas pelas aproximações de B-spline e Bézier, os autores utilizaram redes neurais adversariais generativas (GAN) acopladas a um modelo gerador de B-spline.

Os aerofólios gerados tendem a ser muito mais suaves e mais práticos usando este modelo de geração. A Figura 10 abaixo demonstra amostras de geometrias geradas pelo método de B-spline convencional (Figura 12a) e utilizando a B-spline GAN (Figura 12b).

Figura 12 - Geração de aerofólios de maneira randomizada e utilizando B-spline GAN. Em (a) as geometrias geradas arbitrariamente com curvas b-spline. Em (b), aerofólios gerados pelo modelo b-spline GAN.



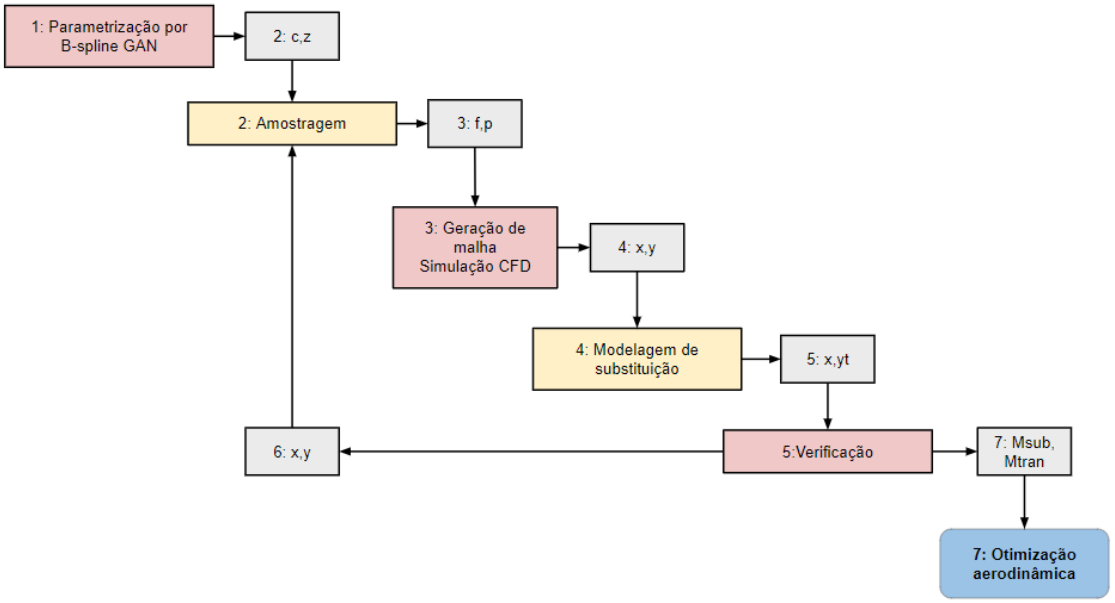
Fonte: Du et al. (2021).

Na Figura 12a, observa-se aerofólios gerados arbitrariamente utilizando curvas B-spline de décima oitava ordem com 18 pontos de controle na superfície do aerofólio. Na Figura 12b observa-se modelos gerados utilizando-se o modelo BSpline-GAN. Foram utilizadas curvas B-spline de décima oitava ordem na última camada do gerador (DU et al., 2021).

Para o estudo, foram consideradas variáveis amplas nas condições de escoamento, tais como número de Mach, número de Reynolds e ângulo de ataque.

Os autores utilizaram dois modelos de MLP para aprendizado dos coeficientes de sustentação e arrasto, além de treinar uma rede neural recorrente (RNN) para o cálculo da distribuição do coeficiente de pressão, que provém informação valiosa para a análise e interpretação dos resultados. O esquema geral do algoritmo é demonstrado na Figura 13.

Figura 13 - Esquema geral do algoritmo de otimização usado em Du et. al (2021)



Fonte: Du et al. (2021).

Inicialmente, na etapa 1 da Figura 13, o modelo B-Spline GAN é treinado. As variáveis de saída são as variáveis latentes c e ruído z do modelo. As variáveis de ruído adicionam variação à população, enquanto as latentes representam características intrínsecas à geometria. A rede é treinada utilizando-se o banco de dados de aerofólios UIC (LI et al., 2019). Na etapa 2, a rede é amostrada, gerando uma população considerando variáveis geométricas e condições de voo diversas, representa as coordenadas do aerofólio gerado e p as condições de contorno. Na etapa 3, a simulação é executada. A variável x representa as variáveis de design e y os resultados de coeficientes de sustentação, arrasto e pressão. Na etapa 4, a rede neural é treinada. $\widehat{y}$ representa os coeficientes estimados pela rede. Nas etapas 5 e 6, o modelo é verificado quanto ao erro gerado e julgamento se o erro é satisfatório. Após, as amostras subsônicas e transônicas são divididas em M_{sub} e M_{tran}, respectivamente. Na etapa 7, uma vez que os modelos apresentam correlação satisfatória, a otimização aerodinâmica é realizada através do otimizador SNOPT (GILL et al., 2005), através da interface pyOptSparse (Wu et al., 2020).

A B-Spline utilizada para a parametrização do aerofólio segue nas Equações 9 e 10 abaixo (DU et al., 2021):

$$B(u) = \sum_{i=0}^{n} N_{i,k}(u) p_i \quad (9)$$

$$N_{i,1}(t) = 1, \; se \; u_i \leq u \leq u_{i+1} \text{ ou } N_{i,1}(t) = 0, \text{ senão}$$

$$N_{i,k} = \frac{u - u_i}{u_{i+k-1} - u_i} N_{i,k-1}(u) + \frac{u_{i+k} - u}{u_{i+k} - u_{i+1}} N_{i+1,k-1(u)} \quad (10)$$

em que *K* é a ordem da curva polinomial da curva, *Nik* são as funções base e *Pi* contém as coordenadas *x* e *y* do enésimo ponto de controle no sistema de coordenadas global.

As coordenadas *X* dos pontos de controle foram mantidas, enquanto *Y* foi variado para gerar os aerofólios. Foram construídas duas B-splines distintas, uma para cada seção do aerofólio. Os pontos da extremidade são coincidentes.

As variáveis independentes do espaço de inferência foram geradas utilizando-se hipercubo latino (LOH, 1996) e variáveis dependentes da cópula gaussiana (LI, 2000).

Para a geração da malha de cada tratamento foi utilizado o software pyHyp (An et al., 2021), além do software ADflow (MADER et al., 2020) como CFD.

Quanto aos treinamentos das redes neurais, os autores utilizaram a técnica de mistura de especialistas, que consiste em utilizar uma combinação de diversas redes neurais treinadas com frações independentes da população de treinamento. De acordo com a literatura, essa técnica traz melhores resultados do que treinar uma única rede (KUNCHEVA, 2004).

Para as MLPs, os autores normalizaram os dados de entrada, utilizaram uma rede de duas camadas com função de ativação Leaky ReLU e o otimizador Adam para a descida do gradiente.

Para as RNNs, os autores utilizaram o modelo LSTM para evitar o problema de dissipação do gradiente (HOCHREITER, 1997).

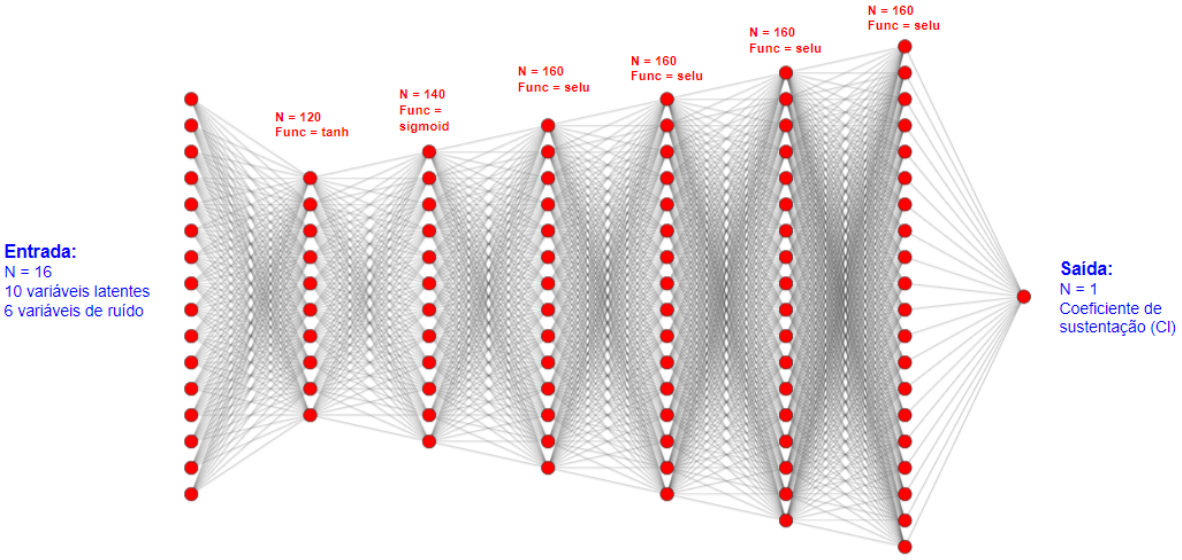
Para o treinamento da GAN, os autores usaram um banco de dados contendo a geometria de 1.552 aerofólios. Cada curva B-spline é de décima oitava ordem e os dois pontos extremos fixos. O erro de utilizar as B-splines para regressão dos aerofólios do banco de dados foi de 0,3%.

A BSpline GAN utilizou 16 variáveis latentes e 10 variáveis de ruído, que foram selecionadas por obter erro baixo na estimativa da espessura dos aerofólios.

A vantagem do uso de B-splines sobre as curvas Bézier é que o modelo é uma generalização das curvas Bézier e independente do número de pontos de controle, além de possuírem melhor controle da geometria.

Para a previsão dos coeficientes de desempenho, Cl e Cd, os autores utilizaram arquiteturas muito similares, redes completamente conectadas (MLPs), realizando otimização do número de neurônios e camadas. A Figura 14 ilustra a arquitetura utilizada para estimativa do coeficiente de sustentação (Cl). Baseado na variação das 26 variáveis de entrada do modelo, os autores geraram um total de 45.696 pontos de treinamento, 330 de validação e 331 de teste, considerando o regime subsônico.

Figura 14 - Rede MLP para previsão de coeficiente de sustentação



Fonte: Do autor, baseado em arquitetura de Du et al. (2021)

De maneira geral, o autor obteve erros máximos relativos entre as previsões do modelo e simulação de CFD de 4,6% para o Cl, 2,87% para o Cd e 11,72% para o coeficiente de pressão, com precisão melhor no regime subsônico. Os autores entenderam que o erro obtido é aceitável para a necessidade da aplicação.

Em resumo, utilizando-se o modelo B-spline GAN para a geração de uma população similar a um banco real, o autor obteve sucesso em treinar modelos de substituição capazes de representar os três coeficientes com erro aceitável e agilizar estimativas e otimizações para aplicações futuras.

Em complemento aos modelos de substituição tradicionais, no qual a variável resposta é uma variável não estruturada, esforços mais recentes têm se concentrado em, além de prever a própria variável de desempenho, estimar os campos de pressão e velocidade gerada em determinada estrutura, tendo a capacidade de fato de atuar em substituição à análise de CFD. As principais vantagens deste tipo de abordagem é que, além de propiciar melhores resultados na própria previsão das variáveis de desempenho, permite uma análise detalhada dos resultados obtidos, além de permitir a confirmação e aderência à física do problema.

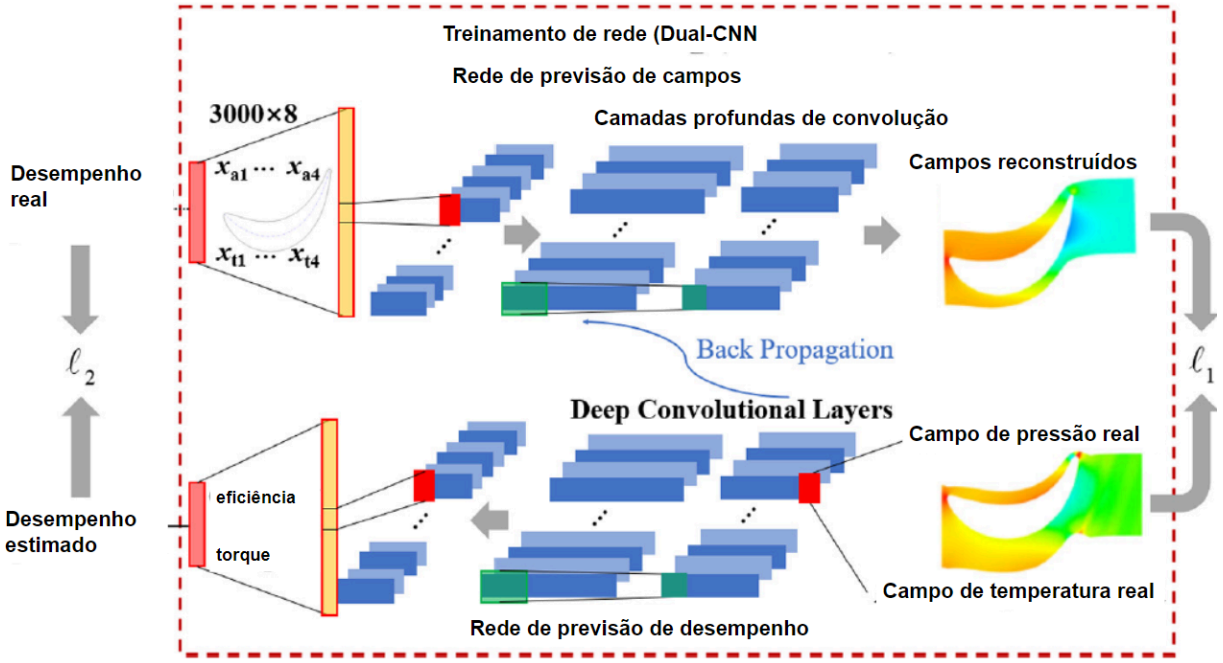
Em Wang et al. (2021) uma arquitetura relacionada ao descrito no parágrafo anterior foi desenvolvida, denominada *Dual-CNN*. A principal diferença foi que, ao invés do modelo ser treinado para a previsão de desempenho, a *Dual-CNN* foi treinada para reconstruir os campos de pressão e temperatura e prever a performance para perfis múltiplos.

Em termos de eficácia, o modelo apresentou performance superior aos modelos de krigagem (KRIGE et al., 1951) e MLP convencional (ROSENBLATT, 1961). O autor ressalta que o treinamento visando a reconstrução dos campos de pressão e temperatura aumentam a acuracidade na previsão de eficiência e torque. A rede *Dual-CNN* expande as funções tradicionais dos modelos de substituição e aplica modelos de reconstrução dos campos de pressão e temperatura para superar a inabilidade dos modelos de substituição tradicionais.

Os campos reconstruídos provêm informações para interpretar os mecanismos de mudança de performance aerodinâmica sem a necessidade de simulação via CFD. Outra vantagem está relacionada aos métodos de otimização empregados na otimização geométrica. Enquanto modelos de substituição tradicionais têm seus modelos de otimização baseados em métodos heurísticos, tais como algoritmos genéticos, recozimento simulado (KIRKPATRICK et al., 1983), etc., o método proposto pode aplicar otimização baseada em gradientes.

O esquema geral do modelo proposto é apresentado na Figura 15. O modelo *Dual-CNN* é composto de uma rede convolucional focada na previsão dos campos de pressão e temperatura, enquanto a outra rede é baseada na previsão das variáveis de performance.

Figura 15 - Esquema geral da arquitetura dual-CNN proposta



Fonte: Wang et al. (2021).

Os dados de entrada da CNN de previsão dos campos de pressão é uma matriz formada por 8 pontos de controle Bézier, relacionados a um determinado perfil aerodinâmico.

Após o processo de treinamento da CNN, a variável resposta são os campos de pressão e temperatura de 512 pontos na interseção entre a superfície do perfil e o centro.

Já as variáveis de entrada da CNN de previsão de performance são os valores reais de temperatura e pressão, com informações de localização. A variável resposta desta rede são a eficiência isentrópica e o torque.

A variável L1 representa a função de perda da rede de previsão de campos, enquanto L2 representa a função de perda da rede de previsão de desempenho. Ambas estão ilustradas na figura 13.

A estrutura da rede *Dual-CNN* é mostrada na Tabela 1 abaixo. Durante o treinamento foi utilizado um *batch size* de 32.

Tabela 1 - Estrutura de rede Dual-CNN

Rede de previsão de campos		**Rede de previsão de desempenho**	
Tipo de camada	*Tamanho de saída do tensor*	*Tipo de camada*	*Tamanho de saída tensor*
Entrada	32 x 8 (parâmetros de perfil)	Entrada	32 x x 512 (parâmetros de campo)
Convolução 1d x 4	32 x 1 x 2048	Convolução 1d	32 x 64 x 256
Convolução 1d x 4	32 x 128 x 16	Convolução 1d	32 x 128 x 128
Convolução 1d x 4	32 x 128 x 32	Convolução 1d	32 x 256 x 64
Convolução 1d x 4	32 x 128 x 64	Convolução 1d	32 x 512 x 32
Convolução 1d x 4	32 x 128 x 128	Convolução 1d	32 x 1024 x 16
Convolução 1d x 4	32 x 128 x 256	Convolução 1d	32 x 2048 x 8
Convolução 1d x 4	32 x 128 x 512	Convolução 1d	32 x 4096 x 4
Saída Convolução 1d	32 x 2 x 512	*Average pooling* 1d	32 x 4096 x 1
		Linear	32 x 512
		Saída	32 x 2

Fonte: Wang et al. (2021)

Na rede de previsão dos campos de pressão, cada camada contém tamanho de *kernel* = 3, *padding* = 1, *stride* = 1 e 128 canais. A função de ativação de cada camada é a *Leaky ReLU*. Já na rede de previsão de performance, cada camada de convolução contém *kernel* = 3, *padding* = 1, *stride* = 2 e ReLU de função de ativação.

Quanto ao processo de otimização, a rede estabelecida é utilizada para obter os parâmetros de performance de cada perfil. A simulação multi-objetiva foi conduzida utilizando um algoritmo baseado em gradiente, utilizado em estudos de aprendizado profundo (Wang et al., 2021). No artigo, os autores comparam a

eficácia do modelo com um modelo tradicional de MLP e um modelo GPR (processo de regressão Gaussiano).

Para os dados de entrada, o perfil da hélice foi representado por duas dimensões, que ajudam a reduzir o número de variáveis de amostragem, o que facilita a otimização. O processo de coleta de dados utilizou a estratégia de hipercubo latino, gerando 3.000 perfis de lâmina. O software CFX 15.0 (ANSYS, 2006) foi utilizado nas simulações. Os campos de pressão e performance aerodinâmica foram obtidos resolvendo as equações de fluidos compressíveis com regime permanente de Navier-Stokes (MALALASEKERA,1995) para o domínio 3D da turbina. Os dados de campo são as informações de pressão e temperatura na interseção entre a superfície e centro da lâmina, combinado com as coordenadas x e y de cada ponto. Os dados são interpolados para formar uma matriz 3.000 x 512 x 2, com pressão e temperatura utilizados para o treinamento da rede de previsão de performance. Os dados de performance e torque são interpolados para formar uma matriz de saída 3.000 x 2.

O perfil é parametrizado utilizando curvas Bézier. A curva B(t) e a função primária de uma curva de ordem N podem ser expressos como segue na Equação 11 (WANG et al., 2021):

$$B(t) = \sum_{i=0}^{n} C_{n,i}(t) X_i \tag{11}$$

$$C_{n,i}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i},$$

em que *B* representa a curva Bézier utilizada, *n* é a ordem da curva que se deseja utilizar, enquanto que $C_{n,i}(t)$ representa a função primária da curva Bézier de ordem *n*.

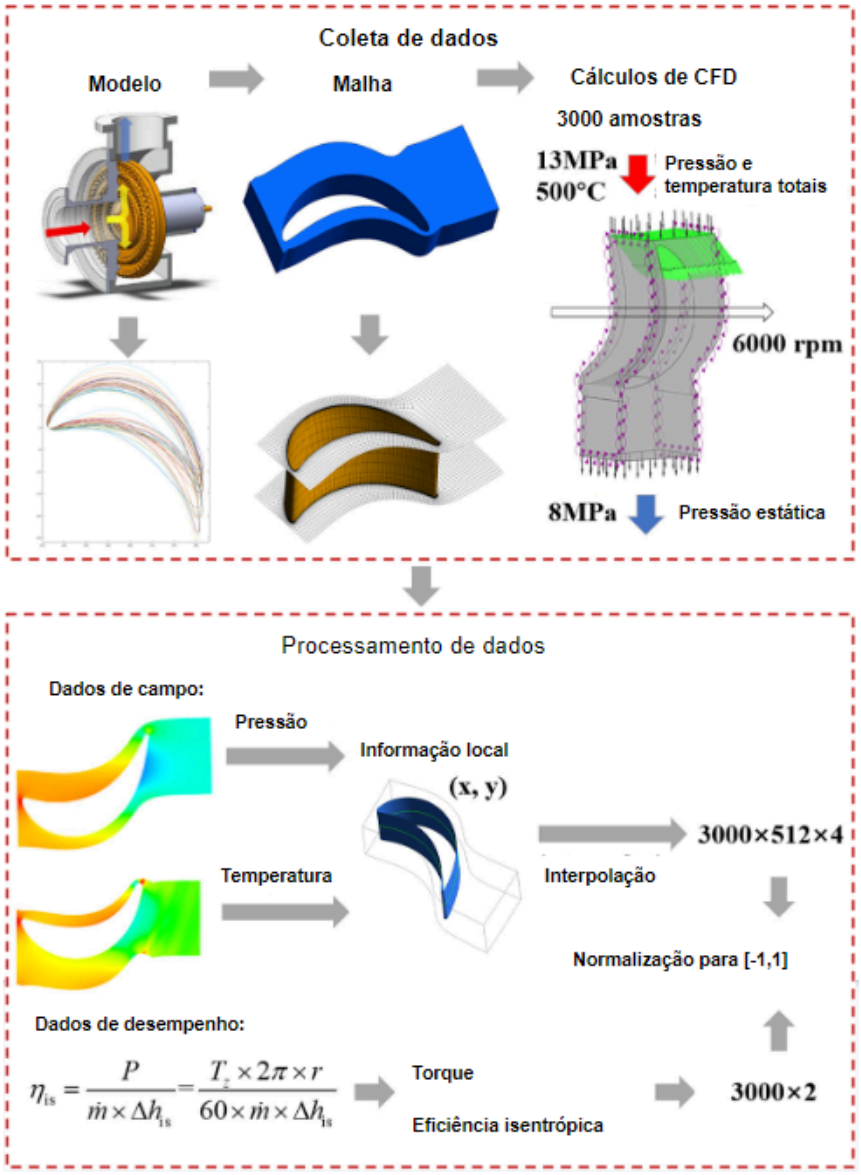
Especialmente, o perfil é definido como a distribuição entre espessura e ângulo da lâmina. Uma curva Bézier cúbica é adotada para quatro pontos de controle, e a equação do perfil pode ser escrita como na Equação 12 (WANG et al., 2021):

$$B(t) = \sum_{i=1}^{4} C_{3,1}(t) xi = X_1(1-t)^3 + 3X_2 t(1-t)^2 + \ldots$$

$$\ldots 3Xt^2(1-t) + X_4 t^3, t \in [0,1] \tag{12}$$

em que x_i representa um ponto de controle e *B* corresponde às curvas de distribuição calculadas para os pontos de controle. Para cada perfil, a distribuição de ângulo da lâmina e espessura são obtidos para os 8 pontos de controle. Posteriormente, o modelo do domínio do fluido pode ser completo. O fluxograma de preparação dos dados de entrada pode ser observado na Figura 16.

Figura 16 - Fluxo de pré processamento dos dados de entrada



Fonte: Wang et al. (2021).

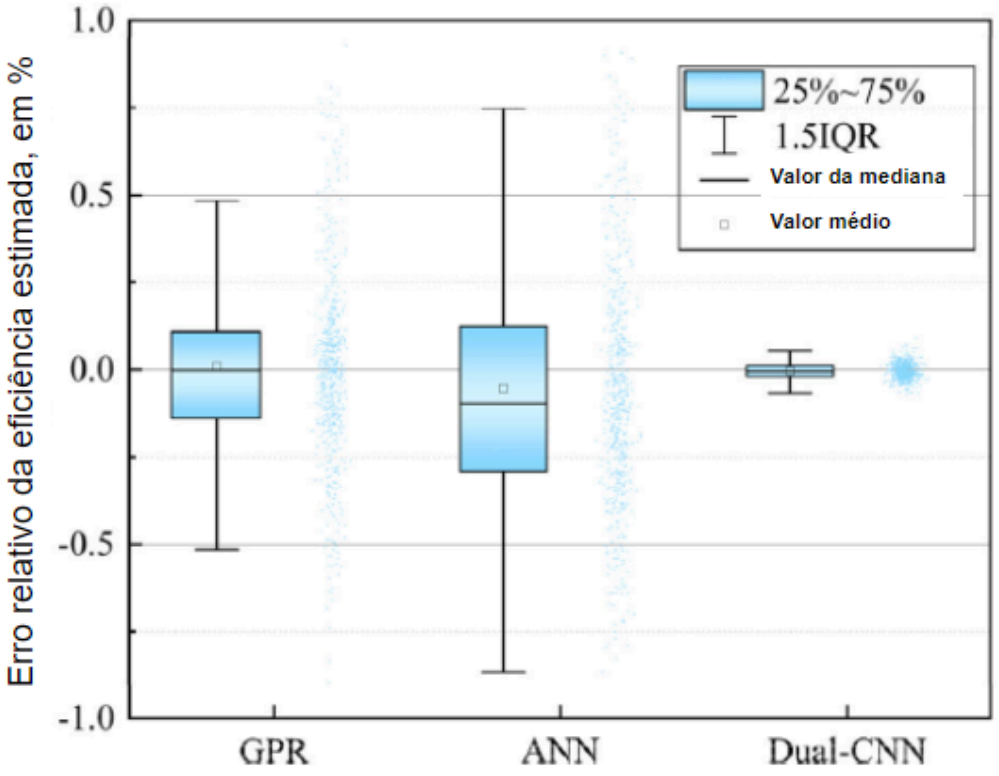
Na Figura 16, η_{is} representa eficiência isentrópica, *P* representa potência, *m* representa o fluxo mássico, Δh_{is} é a variação de coluna manométrica, *T* representa o torque e *r* representa o raio da pá.

Quanto à resolução das equações via CFD, o software CFX utiliza a equação RANS (*Reynolds Averaged NAvier Stokes*) (MALALASEKERA, 1995) para estimar o escoamento turbulento e o método de volumes finitos é adotado na discretização. O modelo K-Omega SST (*Shear stress transport*) (MALALASEKERA, 1995) foi empregado para representar a turbulência. As paredes sólidas adotam condições de contorno adiabático e sem escorregamento. Foram feitos estudos de independência de malha e utilizados elementos tipo “O” (MALALASEKERA, 1995) próximo ao perfil aerodinâmico para se obter resultados de alta qualidade.

Quanto à rede de previsão de performance, o máximo erro relativo entre as previsões foi de menos de 0,5%.

A comparação de erro relativo entre os três métodos de treinamento é mostrada na Figura 17, onde 1.5 IQR representa 1,5 vezes o intervalo interquartil dos dados.

Figura 17 - Erro relativo da eficiência estimada



Fonte: Wang et al. (2021).

Para a aplicação em questão, os autores consideraram que erros entre 1-2% são aceitáveis. O modelo *Dual-CNN* foi utilizado na otimização por ser o mais acurado.

De modo geral, os autores obtiveram sucesso em gerar um modelo de substituição que obteve erro relativo muito baixo, e o aprendizado dos campos de pressão foram importantes para estimar as variáveis de performance e obter a frente de Pareto (FIESLER et al., 2020), entre as variáveis que se desejava otimizar.

Ainda, na abordagem de aplicação de redes convolucionais nos campos de pressão, em Abbas et al. (2021), os autores desenvolveram um esquema computacional baseado na previsão de campos de pressão, temperatura, etc. na superfície de uma geometria, assim como parâmetros escalares gerais sobre qualquer geometria de entrada tridimensional. O modelo também não requer pré-processamento da geometria de entrada e supera o estado da arte em acuracidade de previsão.

O modelo proposto foi baseado em uma rede neural convolucional geométrica (GCNN) (FEY et al., 2018). A rede é capaz de aprender características da superfície 3D diretamente e prever tanto os campos de escoamento quanto quatro variáveis escalares.

No estudo, os autores utilizaram dois bancos de dados principais. O primeiro banco de dados consiste de 700 geometrias de navio, geradas pelos autores, considerando geometria completamente parametrizada. De maneira geral, cada geometria foi composta de 5 tipos básicos de casco e 8 diferentes tipos de deck. Os modelos foram gerados automaticamente e a simulação de CFD executada no software OpenFOAM (JASAK e JEMCOV, 2007).

Já para a comparação dos modelos gerados com a literatura, os autores utilizaram o banco de dados "*car dataset*", que consiste nas soluções escalares dos campos de pressão de 889 geometrias de carro, geradas em Umetani e Bickel (2018), a partir da categoria 'carro' do banco de dados ShapeNet (CHANG et al., 2015).

A rede neural utilizada consiste em dois blocos principais, nomeados bloco PointNet (QI et al., 2017) e o bloco de convolução geométrica. A arquitetura PointNET++ foi proposta inicialmente por Qi et al. (2016) e introduz uma estrutura hierárquica, permitindo à rede capturar características em diferentes escalas. Contudo, o refinamento excessivo gera custos computacionais altos para capturar

detalhes finos, o que inibe seu uso em bancos de dados contendo muitos pontos. Desta maneira, os autores optaram por utilizar uma rede PointNet convencional, que mantém a capacidade de aprender características em maior escala, mas com eficácia limitada para aprender estruturas locais.

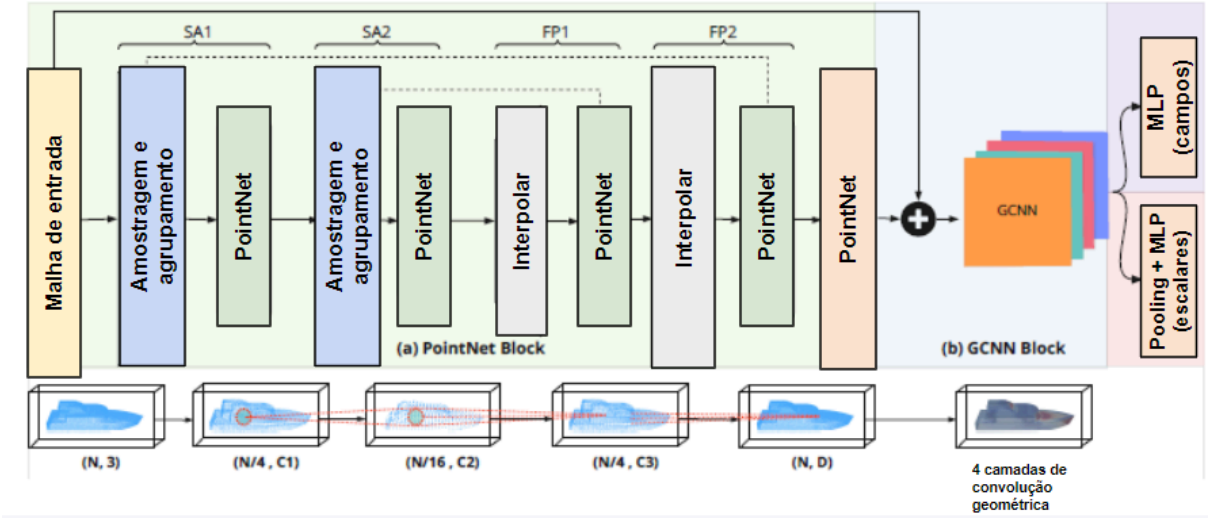
O segundo bloco, da convolução geométrica, supera essa limitação, permitindo ao modelo o aprendizado de detalhes finos. Comparado com a representação em pontos da rede PointNet, a rede é baseada na leitura de malhas, possuindo conectividade entre pontos vizinhos. Desta maneira, ela é mais eficiente em capturar detalhes finos. O uso em paralelo dos dois blocos, de acordo com os autores, se deve ao fato de que a informação extraída da nuvem de pontos auxilia a rede GCNN em melhorar o aprendizado em malhas grandes.

O bloco PointNet tem como entrada um grupo de pontos, representados como vetores de posição (x,y,z). A rede possui dois blocos de Abstração (SA1 e SA2) e, posteriormente, dois blocos de propagação (FP1 e FP2), que agrupam pontos em N1 = 512 e N2 = 128 regiões locais, respectivamente. Na fase de amostragem e agrupamento é utilizado o método do ponto mais distante para amostrar centróides e o método de ‘ball query’ (PEDRINI et al., 2008) para agrupar pontos. Esses dois níveis extraem C1 = 128 e C2 = 256 características dimensionais de cada região local. Os dados da última camada de abstração são passados através de dois módulos de propagação sucessivos. Após as camadas de propagação, a rede possui camadas completamente conectadas para a extração de um vetor de 8 características para cada ponto.

A entrada do bloco de convolução geométrica é a malha contendo os pontos e as 8 dimensões adicionais extraídas pela PointNet. A rede GCNN aplica o conceito de Spline-CNN (FEY et al., 2018) com 4 camadas convolucionais, *Spline Convolution* (1, 16), *Spline Convolution* (16, 32), 2 x *Spline Convolution* (32, 64), com tamanho de *kernel* 5. A primeira coordenada de cada camada de convolução com spline representa o número de mapas de características de entrada e a segunda o número de mapas de característica da saída. Para as previsões de variáveis escalares, um modelo de regressão composto por 2 camadas completamente conectadas com 64 e 32 neurônios, seguido por *Max pooling* mais uma camada de 32 neurônios, todos com função de ativação ReLU.

Uma ilustração da arquitetura da rede neural proposta por Abbas et al. (2021) está exibida na Figura 18.

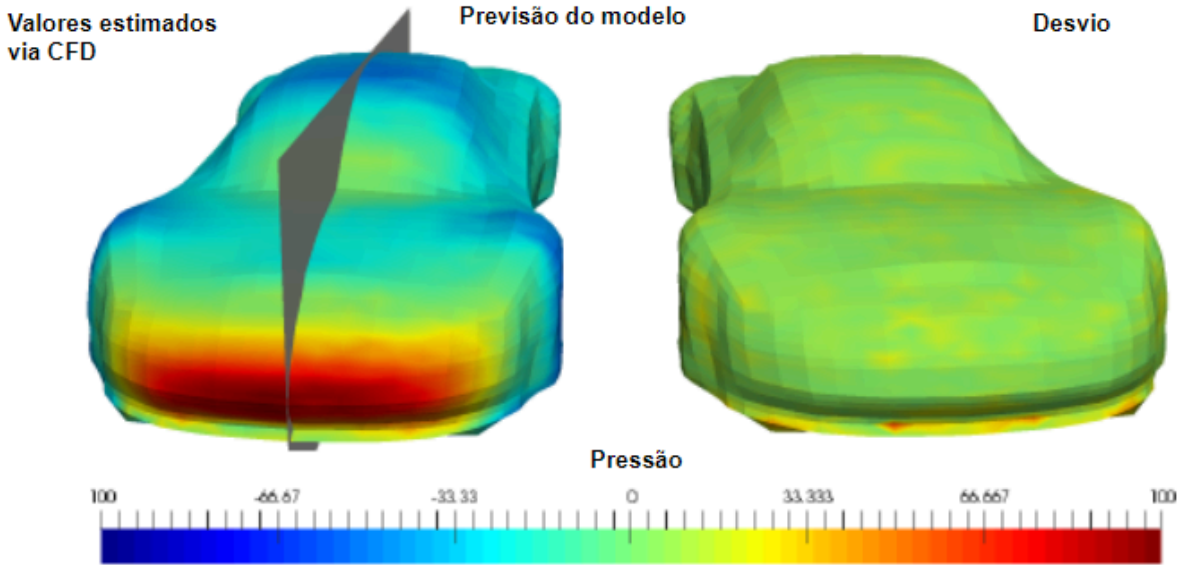
Figura 18 - Arquitetura para estimativa de campos de pressão e variáveis escalares



Fonte: Abbas et al. (2021).

Para o treinamento com o banco de dados '*car dataset*', os dados obtidos mostraram um desvio padrão e erros absolutos médios superiores à literatura para a previsão do coeficiente de arrasto. A Figura 19 demonstra os resultados para a previsão de pressão na superfície.

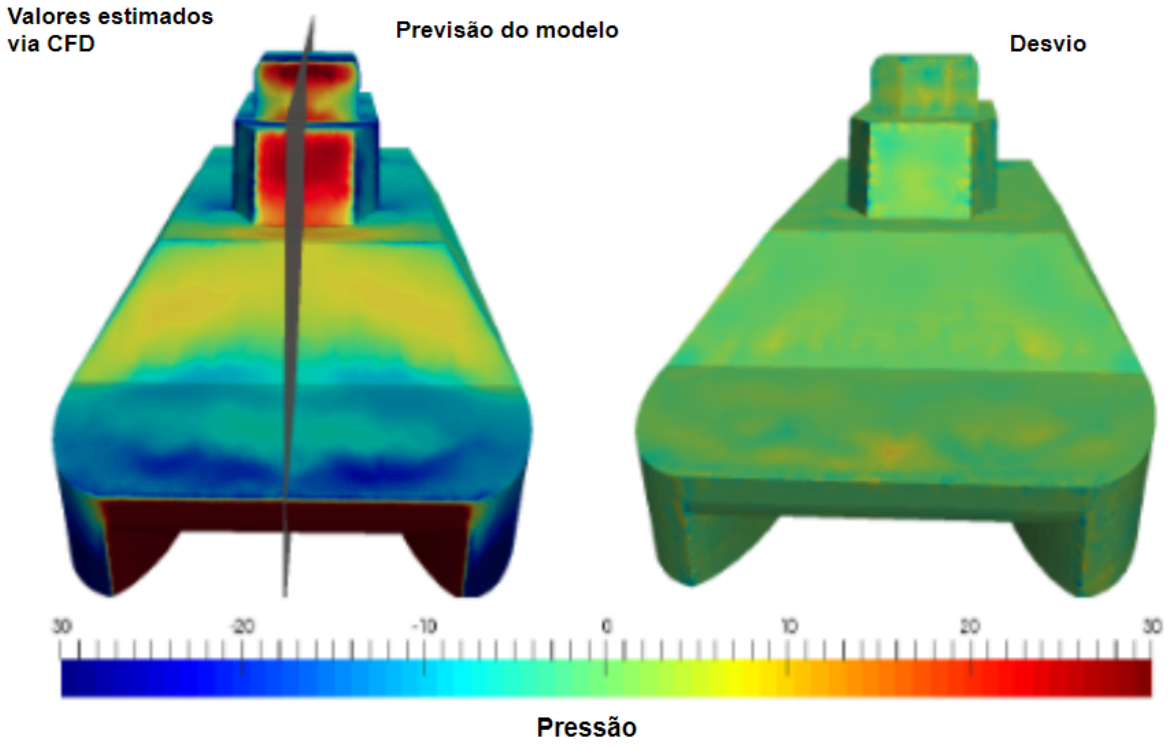
Figura 19 - Previsão dos campos de pressão estimados pelo CFD em comparação aos campos estimados pela rede neural



Fonte: Abbas et al. (2021).

O banco de dados com as geometrias dos barcos é bem mais complexo, apresentando cerca de 20 vezes mais vértices que os carros. Além disso, o banco possui resultados de força nas duas direções. Os autores realizaram o treinamento considerando apenas a PointNet, a GCNN e a arquitetura mista com os dois blocos. Os resultados obtidos demonstram que o erro médio quadrático e erro absoluto percentuais foram menores com a arquitetura mista. A Figura 20 demonstra os dados qualitativos obtidos em um dos exemplos estimando os campos de pressão para uma configuração de barco.

Figura 20 - Campos de pressão estimado para modelo 3D de barco



Fonte: Abbas et al. (2021).

Os erros percentuais obtidos foram na faixa de 4% a 6% para as duas direções de força, similares aos obtidos para as direções x e y, que são equiparáveis aos próprios erros experimentais deste tipo de aplicação.

Em linhas gerais, para o banco de dados "*car dataset*", os autores conseguiram com o modelo proposto reduzir o erro encontrado com o estado da arte para a aplicação. No caso de resistência do vento para os barcos, o menor erro encontrado foi na arquitetura proposta, com performance superior aos modelos PointNet e GCNN independentes.

De maneira geral, diversos autores obtiveram resultados relevantes, aliando arquiteturas de aprendizado supervisionado e generativo, em diversas geometrias diferentes. A Tabela 2 destaca a contribuição principal de cada trabalho abordado nesta seção.

Tabela 2 - Resumo de trabalhos abordados na seção

Autor	Descrição
Lallonde et al. (2021)	Comparação de arquitetura de rede MLP, rede LSTM e rede convolucional na previsão de desempenho de turbinas eólicas. A rede convolucional obteve maior êxito que as demais, com o menor erro em comparação às demais arquiteturas.
Du et al. (2021)	Utilização de rede generativa B-spline GAN e MLP para otimização de aerofólios aeronáuticos, baseado em perfis NACA. Rede generativa foi capaz de otimizar a geração de população de treinamento considerando banco de aerofólios fisicamente viável.
Wang et al. (2021)	Uso de rede convolucional dupla (*Dual-CNN*) para geração de modelo de substituição de performance aerodinâmica de perfis de turbina. Redes convolucionais treinadas em série para estimar tanto campos vetoriais quanto resultados escalares. Resultado dos escalares foi potencializado pela predição dos campos vetoriais.
Abbas et al. (2021)	Utilização de rede convolucional considerando aplicação de spline-CNN e PointNet para treinamento de campos vetoriais provenientes de simulação fluidodinâmica. Resultados apresentaram aderência com os resultados obtidos via simulação, tanto para um banco de treinamento de embarcações quanto para carros.

3.5. Considerações finais da análise bibliográfica

De maneira geral a Seção 3.1 deste capítulo teve como objetivo sumarizar o objetivo e as vantagens de se utilizar a fluidodinâmica computacional (CFD) para resolução de problemas relacionados ao escoamento de fluidos, as equações regentes da física deste problema específico e a mecânica de concepção de uma simulação de fluidodinâmica computacional. Além disso, foram detalhadas as equações regentes de maneira geral deste tipo de fenômeno e mencionado de modo sucinto o processo de discretização que permite a solução do problema. Foi destacado que, apesar de trazer diversos benefícios na concepção de um modelo, as simulações usualmente são longas e demandam um esforço computacional

grande, o que a torna por vezes impeditiva para aplicação de um algoritmo de otimização de maneira direta.

Nas Seções 3.2 e 3.3 foram detalhadas 2 tipos principais de arquitetura de aprendizado de máquina, as redes do tipo multilayer perceptron (MLPs) e as redes neurais convolucionais (CNNs). Este detalhamento é importante pois baseia os trabalhos nos quais aprendizado de máquina foi empregado para treinar modelos, denominados na literatura de modelos de substituição, a partir de resultados gerados via CFD.

Na Seção 3.4 foram abordados trabalhos em que modelos de fluidodinâmica computacional foram empregados para a geração de modelos de substituição. Estes modelos abrangem a utilização destas redes para geração direta da variável de desempenho e também para modelos que são capazes de aprender os próprios campos da simulação. Além disso, foi detalhado um trabalho onde uma rede generativa foi utilizada para geração dos exemplos de treinamento do modelo de substituição, a partir de um banco de dados já existente.

4. MATERIAIS E MÉTODOS

Este capítulo tem como objetivo detalhar a proposta de etapas de procedimento experimental que se pretende executar para a obtenção do algoritmo de otimização.

Inicialmente, é necessária a criação de um banco de dados representativo, uma vez que a literatura apresenta poucos bancos de dados que possuem contexto parecido com a aplicação em questão.

A Seção 4.1 abordará a proposta de parametrização do modelo de dispensador de gaveta, incluindo as variáveis independentes escolhidas para a criação do modelo. Serão incluídos também detalhes sobre a proposta de plano de geração da população de treinamento, baseado em um experimento fatorial completo, complementado por um experimento de design composto central e as razões para tal.

Na Seção 4.2 serão detalhados os fluxos de definição da simulação no software de simulação *Fluent* até a execução da simulação (FLUENT, 2024).

Na Seção 4.3 será detalhada a proposta de pós-processamento dos dados resultantes da simulação, de maneira a utilizá-los como dados de treinamento para um modelo de aprendizado supervisionado. Então, serão abordados a estrutura dos dados de entrada de treinamento e os dados de saída.

Na Seção 4.4 será detalhada a proposta base da arquitetura da rede neural convolucional (CNN) capaz de utilizar os dados de entrada e saída para treinamento do modelo. Serão detalhados também os experimentos que serão utilizados para os campos de VOF e cisalhamento para a otimização dos hiperparâmetros da rede.

A Seção 4.5 detalha as métricas de desempenho da rede neural treinada, de maneira a verificar a acuracidade do perfil gerado em relação ao banco de treinamento gerado via simulação. Além disso, será descrito como será gerado um modelo de regressão do erro para cada variável independente da população de treinamento, de maneira a encontrar regiões onde o modelo de substituição é mais ou menos eficiente em substituir a simulação.

4.1. Geração de modelo de CAD paramétrico e geração de dados de treinamento

A primeira etapa do processo de simulação é gerar um modelo computacional que, através de uma determinada combinação de variáveis independentes, consiga gerar uma geometria de dispensador em função de um determinado conjunto de variáveis independentes. Desta maneira, um modelo tridimensional do conjunto dispenser simplificado é criado a partir da definição destas variáveis independentes.

A critério de exemplificação, considera-se que esta geometria pode ser representada por três variáveis independentes A, B e C, sendo:

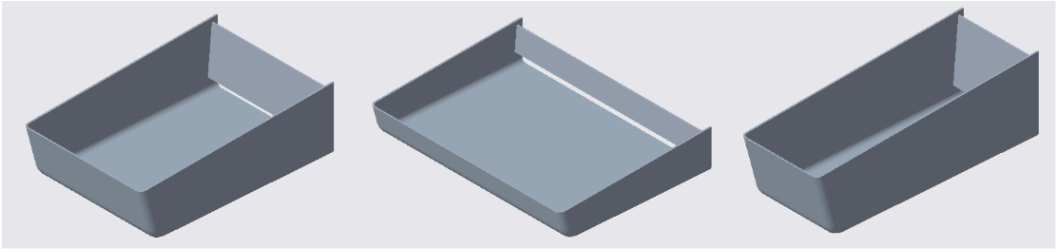
A: Largura da cavidade;

B: Comprimento da cavidade;

C: Altura da cavidade.

Desta maneira, uma vez que o modelo parametrizado é criado, o software de CAD permite a geração de modelos a partir de combinações destas variáveis independentes. As geometrias foram criadas utilizando-se um software de CAD embutido no software de simulação Fluent, denominado *SpaceClaim* (ANSYS, 2024). A Figura 21 ilustra três variantes de cavidade geradas a partir do mesmo modelo paramétrico.

Figura 21 - Exemplos de variantes de cavidade criadas a partir das variáveis A, B e C



Fonte: do autor

O modelo paramétrico gerado foi representado por um total de 15 variáveis independentes, sendo 14 variáveis relacionadas a características geométricas do componente e uma variável relacionada à vazão de entrada de água no sistema. As variáveis estão descritas na tabela 3.

Tabela 3 - Variáveis independentes do modelo

Variável	Componente	Descrição
X1	Cavidade	Largura da cavidade
X2	Cavidade	Comprimento da cavidade
X3	Cavidade	Altura da cavidade
X4	Cavidade	Ângulo da base da cavidade
X5	Cavidade	Ângulo frontal da cavidade
X6	Cavidade	Altura da abertura traseira
X7	Cavidade	Ângulo da parede traseira
X8	Cavidade	Altura entre parede traseira e plano superior
X9	Espalhador	Espaçamento traseiro espalhador
X10	Espalhador	Altura do espalhador
X11	Espalhador	Diâmetro furos do espalhador
X12	Espalhador	Distância lateral dos furos do espalhador
X13	Espalhador	Distância frontal dos furos do espalhador
X14	Espalhador	Largura do canal de entrada
X15	Sistema	Vazão de entrada de água

Uma vez definidas as variáveis amostrais, para cada variável independente foram definidos dois limites operacionais máximo e mínimo, baseado na experiência da aplicação e em componentes já existentes. Para a geração do banco de dados, optou-se por utilizar um experimento do tipo *Design of Experiments* (DOE), onde as 14 variáveis foram variadas independentemente (MONTGOMERY, 2017).

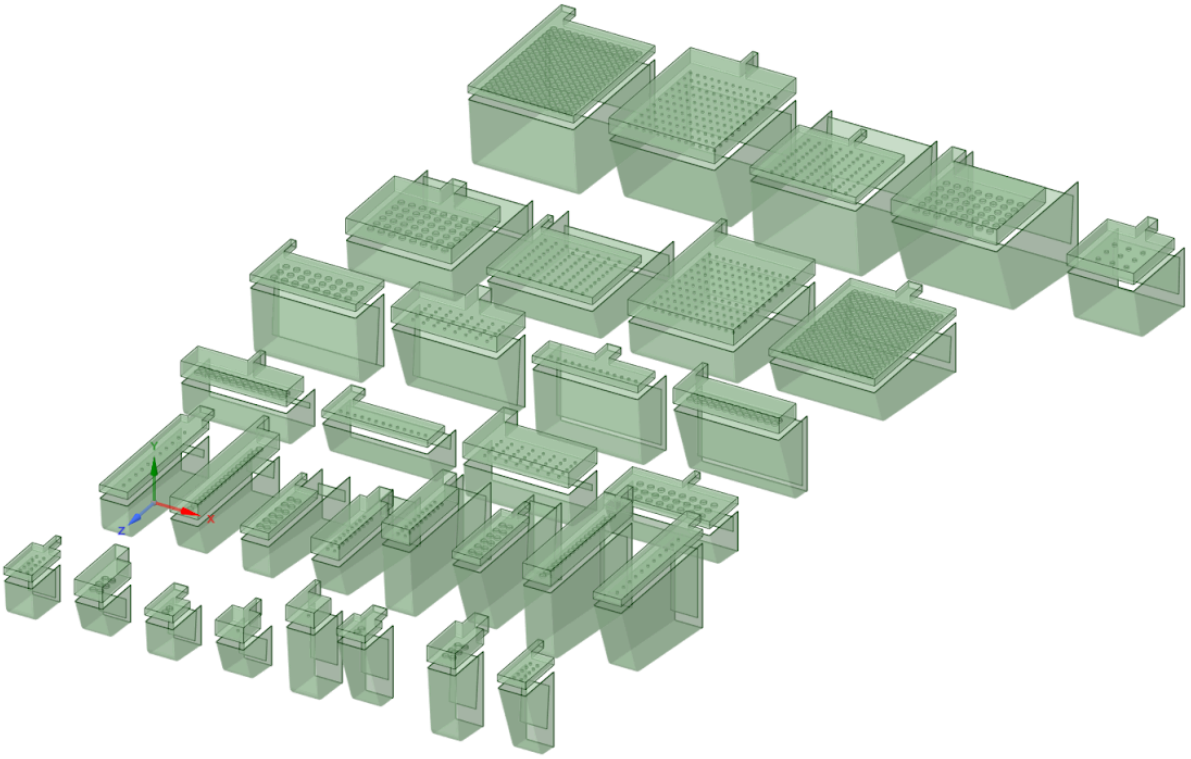
Neste experimento, cada fator, no caso uma variável geométrica, foi definida entre um nível inferior, denominado “-”, e um nível superior, denominado “+”. Desta maneira, 64 tratamentos independentes considerando-se os níveis “-” e “+” foram realizados, considerando-se as 14 variáveis geométricas independentemente. O DOE é um tipo específico de regressão utilizando-se mínimos quadrados onde o

objetivo é calcular uma função de regressão conforme Equação 12 abaixo (MONTGOMERY, 2017):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \beta_{X1X2} X_1 X_2 + \beta_{XnXm} X_n X_m \quad (12)$$

Cada β_i na equação acima representa o coeficiente angular calculado para cada variável independente do modelo regressor, enquanto cada X_i representa o nível daquela variável independente, considerando os valores normalizados entre -1 e 1. Esta função de regressão permite calcular como a mudança de nível de cada fator entre os valores de '-' e '+' afeta o resultado esperado do modelo. O modelo de experimento fatorial é criado de maneira que cada nível de cada fator ocorre um mesmo número de vezes no experimento. No experimento acima, por exemplo, como o experimento possui 64 tratamentos, cada variável possui 32 combinações relacionadas ao nível '-' e 32 combinações relacionadas ao nível '+'. A Figura 22 abaixo mostra uma vista diagramática com 32 das geometrias geradas a partir da definição de níveis '-' e '+' para as variáveis independentes. Nota-se a altíssima variabilidade de geometrias geradas a partir do mesmo código fonte. Entende-se que geometrias futuras a serem desenvolvidas estão contidas dentro deste espaço de inferência (MONTGOMERY, 2017).

Figura 22 - Geometrias geradas a partir do mesmo modelo paramétrico



Fonte: Do autor

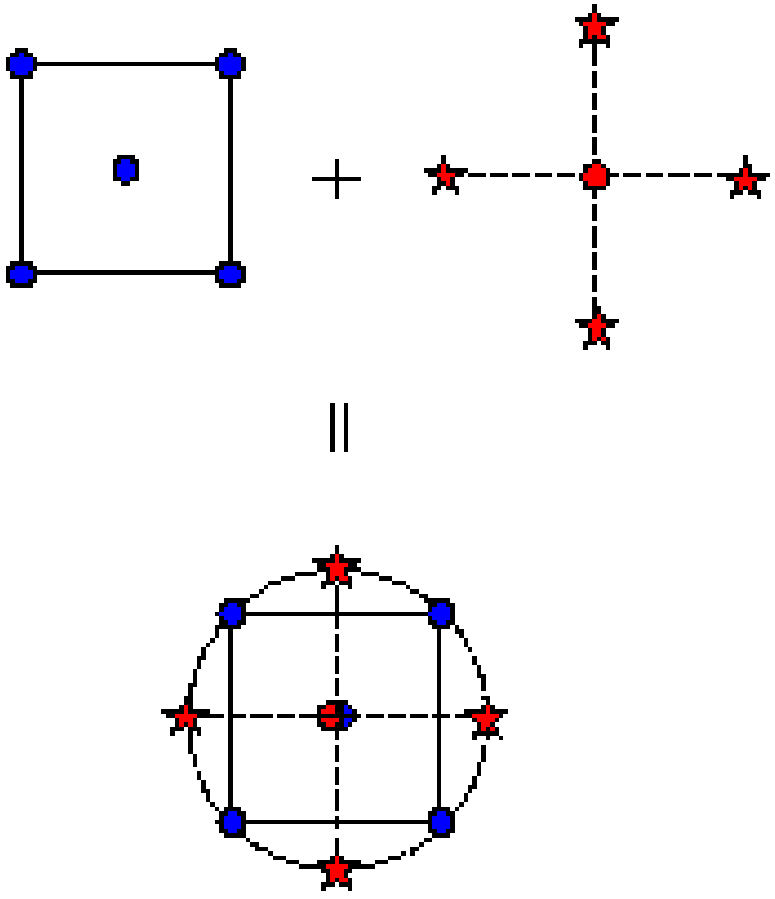
Em função da estruturação do experimento, duas propriedades denominadas balanceamento e ortogonalidade reduzem significativamente os efeitos da denominada multicolinearidade, que é relacionada ao nível de correlação entre duas variáveis independentes em um modelo de regressão. Desta maneira, a dispersão dos erros dos efeitos de cada fator é minimizada e pode-se estimar com mais certeza os fatores de fato significativos (MONTGOMERY, 2017).

O intuito de criar os dados de treinamento desta maneira foi que, uma vez que o treinamento da rede permite estimar o erro da função de perda para cada membro da população, é possível criar um modelo de regressão onde o erro do treinamento *Y* pode ser entendido a partir dos níveis das 15 variáveis independentes do modelo e suas interações principais. Desta maneira, além do erro médio do modelo, é possível estimar em quais regiões a acuracidade do treinamento é melhor ou pior que as demais regiões.

Além do experimento fatorial, outras 60 rodadas foram adicionadas a este bloco. 29 destes pontos se referem a um complemento do experimento fatorial, denominado design de componente central de Box-Wilson (*Central composite design, ou CCD*) (BOX E WILSON, 2017). Este experimento pode ser entendido

conforme ilustrado na Figura 23. Imaginando-se um estudo fatorial de dois níveis, caso apenas dois fatores tivessem sido avaliados, resultaria em uma geometria de quadrado, conforme ilustrado à esquerda da Figura 23. Além destes pontos dos vértices do cubo, em um experimento de design de componente central, os tratamentos adicionais à direita são considerados. Estes tratamentos tratam-se de configurações onde apenas uma variável é alterada enquanto as outras são mantidas no valor central do espaço de inferência destas. Desta maneira, além do efeito das variáveis independentes e interações descritos na Equação 12, é possível estimar os termos quadráticos relacionados a cada um dos graus de liberdade, ou efeitos. Desta maneira, é possível entender o efeito de não linearidade de cada fator, adicionando-se um termo quadrático na regressão (MONTGOMERY, 2017).

Figura 23 - Experimento fatorial complementado por design composto central



Fonte: STANDARD, 2002

Por último, para ajudar no treinamento da rede e melhorar sua generalização, a técnica de hipercubo latino foi aplicada, gerando-se 30 geometrias dentro do espaço de inferência mínimo e máximo de cada variável (LOH, 1996). Desta maneira, o número total de tratamentos do estudo foi de 123 pontos por nível de vazão, considerando-se 64 pontos provenientes do experimento fatorial completo, 29 pontos do experimento de design composto central e 30 pontos de uma amostragem de hipercubo latino para as 14 variáveis. Este experimento foi replicado para blocos de vazão de 3,6 e 9 L/min, totalizando 369 rodadas de simulação.

A árvore de amostragem para cada um dos blocos pode ser vista na Figura 24 abaixo.

Figura 24 - Árvore de amostragem de cada bloco de vazão

<table>
<tr><td>Setup simulação</td><td colspan="33">1</td></tr>
<tr><td>Modelo raiz simulação</td><td colspan="33">1</td></tr>
<tr><td>Método de extração campos</td><td colspan="33">1</td></tr>
<tr><td>Inclinação produto</td><td colspan="33">0° paralelo ao chão (inclinado pra frente 2° e 2° traás)</td></tr>
<tr><td>Temperatura água</td><td colspan="33">25°C (25 e 40)</td></tr>
<tr><td>Vazão de água</td><td colspan="33">6 L/min(3 a 9)</td></tr>
<tr><td>Raio bordas</td><td colspan="33">5 mm</td></tr>
<tr><td>Inclinaçãop frontal</td><td colspan="33">15°</td></tr>
<tr><td>Espessura componentes</td><td colspan="33">1 mm</td></tr>
<tr><td>Distancia parede traseira e parede contenção</td><td colspan="33">10 mm</td></tr>
<tr><td>Altura ombros furos espalhador</td><td colspan="33">2 mm</td></tr>
<tr><td>Espaço de ar entre espalhador e gaveta</td><td colspan="33">10</td></tr>
<tr><td>Margem X</td><td colspan="33">0</td></tr>
<tr><td>Comprimento entrada de água</td><td colspan="33">30 mm</td></tr>
<tr><td>Largura da cavidade</td><td colspan="16">-</td><td>0</td><td colspan="16">+</td></tr>
<tr><td>Comprimento da cavidade</td><td colspan="8">-</td><td colspan="8">+</td><td>0</td><td colspan="8">-</td><td colspan="8">+</td></tr>
<tr><td>Altura da cavidade</td><td colspan="4">-</td><td colspan="4">+</td><td colspan="4">-</td><td colspan="4">+</td><td>0</td><td colspan="4">-</td><td colspan="4">+</td><td colspan="4">-</td><td colspan="4">+</td></tr>
<tr><td>Ângulo da base da cavidade</td><td colspan="2">-</td><td colspan="2">+</td><td colspan="2">-</td><td colspan="2">+</td><td colspan="2">-</td><td colspan="2">+</td><td colspan="2">-</td><td colspan="2">+</td><td>0</td><td colspan="2">-</td><td colspan="2">+</td><td colspan="2">-</td><td colspan="2">+</td><td colspan="2">-</td><td colspan="2">+</td><td colspan="2">-</td><td colspan="2">+</td></tr>
<tr><td>Ângulo frontal da cavidade</td><td>-</td><td>+</td><td>-</td><td>+</td><td>-</td><td>+</td><td>-</td><td>+</td><td>-</td><td>+</td><td>-</td><td>+</td><td>-</td><td>+</td><td>-</td><td>+</td><td>0</td><td>-</td><td>+</td><td>-</td><td>+</td><td>-</td><td>+</td><td>-</td><td>+</td><td>-</td><td>+</td><td>-</td><td>+</td><td>-</td><td>+</td><td>-</td><td>+</td></tr>
<tr><td>Altura da abertura traseira</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>0</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Ângulo da parede traseira</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>0</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Altura entre parede traseira e plano superior</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>0</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Espaçamento traseiro espalhador</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>0</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Altura do espalhador</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>0</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Diâmetro furos do espalhador</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>0</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Distância lateral dos furos do espalhador</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>0</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Distância frontal dos furos do espalhador</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>0</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Largura do canal de entrada</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>0</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Y1: Shear profile</td><td colspan="33"></td></tr>
<tr><td>Y2: VOF profile</td><td colspan="33"></td></tr>
</table>

Fonte: Do autor

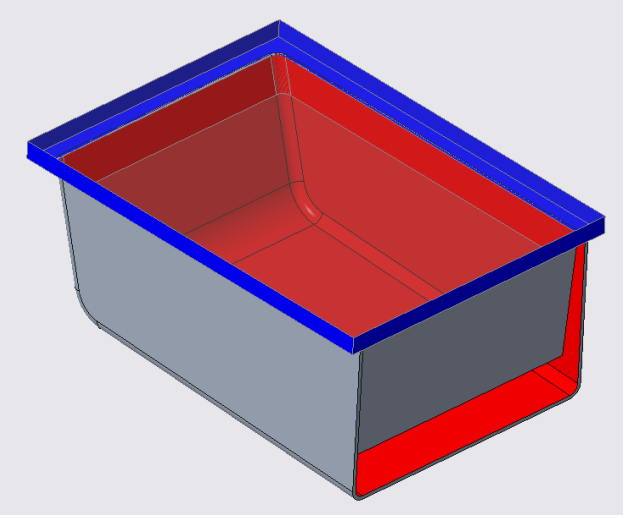
4.2. Pré-processamento e detalhamento de modelo de CFD

Parte fundamental da concepção do modelo de simulação computacional é a geração da malha. Uma vez que a malha é a metodologia de discretização de determinado sistema físico, ela deve seguir determinadas práticas para garantir que o erro obtido seja minimizado durante a discretização. Desta maneira, a malha deve possuir tamanhos de elemento suficientes para representar eficientemente o escoamento. Existem diversas métricas que podem ser aplicadas para a verificação da qualidade da malha, tais quais o número de *Courant* (MALALASEKERA, 1995), que estima o tamanho máximo do elemento considerando-se a velocidade do escoamento do fluido esperada.

Ao mesmo tempo, a malha a ser gerada deve possuir uma estrutura que possa ser mais facilmente pós-processada. Deste modo, a malha foi gerada de maneira estruturada, de modo a ter elementos de tamanho constante em cada parede.

Dentro da malha, cada região do volume tem um elemento associado. Contudo, para a aplicação em questão, apenas os elementos associados às superfícies destacadas em vermelho na Figura 25 foram utilizados no pós-processamento. As superfícies destacadas em vermelho são de interesse porque representam o perfil de cisalhamento e velocidade do fluido nas paredes internas da gaveta, além de indicar qual a fração das paredes de fato são cobertas por água.

Figura 25 - Superfícies a serem usadas no pós-processamento da simulação



Fonte: Do autor

Já a região em azul representa o domínio de ar sobre a gaveta. A presença de água em contato com estas paredes indica que houve transbordamento da cavidade durante a transferência de água do espalhador para a gaveta.

A simulação em questão utilizou a metodologia VOF (*Volume of Fluid*) (HIRT et al., 1981). Este método é amplamente utilizado por reproduzir de maneira eficiente o comportamento de fluidos escoando em uma superfície livre, tais como regiões de interface de material, ondas de choque ou interface entre fluidos e estruturas deformáveis. O problema de dispensamento se aplica a este método de resolução por tratar-se de fato de um escoamento de água sobre uma superfície plástica, sendo todo o volume não ocupado por partículas de água sendo ocupadas por partículas de ar atmosférico.

De maneira simplificada, usualmente, cada elemento da malha utiliza apenas um conjunto de propriedades para representar o estado do fluido. O método, no entanto, introduz a variável VOF, denominada volume fracional. O valor unitário de VOF representaria um elemento completamente preenchido de água, enquanto um valor de VOF igual a 0 significa que aquele elemento está ocupado apenas por ar. Células cujo valor de VOF se encontram entre 0 e 1 representam células em superfícies livres, ou seja, elementos que se encontram na interface entre os dois fluidos (ar e água, na aplicação em questão).

Em simulações de fluidodinâmica computacional é fundamental a realização de estudos relacionados à independência de malha. Uma vez que a simulação se baseia na discretização de um domínio contínuo em domínios simplificados, uma quantidade insuficiente de elementos, especialmente próximo onde há grandes gradientes de velocidade e pressão, como interfaces líquido-sólido ou gás-líquido, denominada camada limite, pode gerar erros significativos nos cálculos do sistema.

O objetivo do estudo de independência de malha é garantir que o tamanho mínimo de elemento é suficiente para representar o comportamento físico do sistema. O estudo é realizado refinando-se sucessivamente a malha, ou seja, reduzindo o tamanho de seus elementos, de maneira a comparar os resultados do domínio mais refinado em relação ao menos refinado. Entende-se que o sistema atingiu independência de malha quando há um erro considerável aceitável entre dois níveis de refino sucessivos. Além do mais, pode-se utilizar o número de *Courant* (MALALASEKERA, 1995) menor que 1. Para o sistema em questão, o estudo de independência de malha foi realizado considerando-se o pior caso entre os

tratamentos de treinamento (maior vazão, menor volume de controle). Este tamanho padrão foi então definido para os demais tratamentos do treinamento.

4.3 Pós-processamento dos dados de simulação

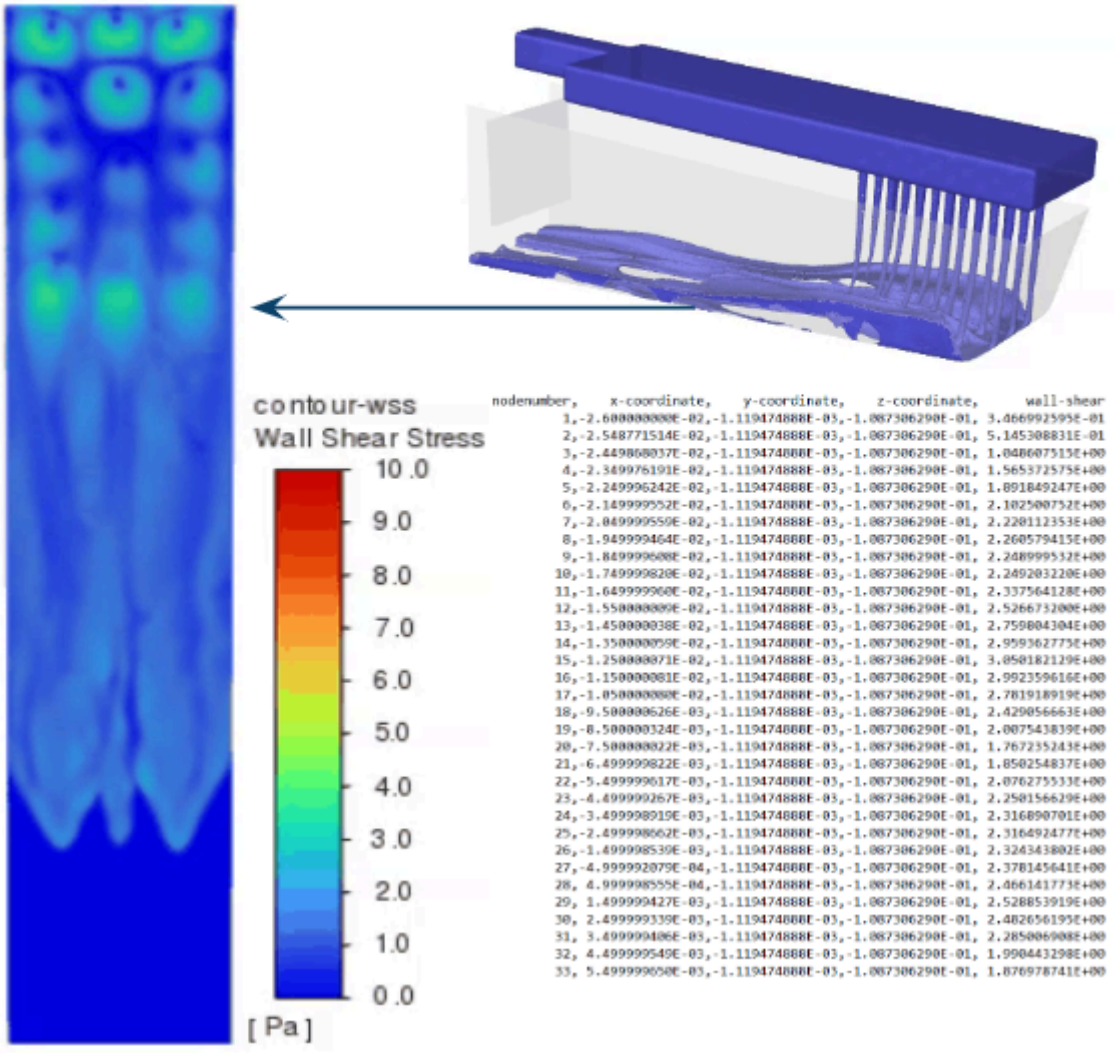
Após o processamento da simulação, cada indivíduo viável da população de treinamento apresenta dois arquivos resultantes com estrutura Nx5, onde N representa o número de elementos nas paredes da malha e os cinco canais representam, respectivamente, o índice de determinado elemento, as posições de cada elemento em relação às coordenadas cartesianas x, y e z e os valores de cisalhamento ou fração volumétrica de cada um.

A Figura 26 descreve de maneira diagramática a sequência de considerações para tradução dos resultados de simulação em um arquivo de treinamento.

Após a conclusão da simulação, o escoamento de água sob a superfície inferior da cavidade gera um determinado perfil de cisalhamento e fração volumétrica sob aquela superfície. Desta maneira, o mapa de calor à esquerda da Figura 26 demonstra o perfil de cisalhamento nesta superfície.

No canto inferior da Figura 26, pode-se verificar uma vista truncada do arquivo de saída de tamanho Nx5, gerado a partir da exportação dos resultados obtidos em cada um dos elementos.

Figura 26 - Processo de extração de perfil de cisalhamento sob superfície de interesse da cavidade.



Fonte: Do autor

A fim de ajustar a dimensão dos arquivos da simulação para os tornar adequados para o processo de treinamento, foi aplicado um processo de conversão do domínio de todas as superfícies para um tamanho padrão de 128 x 128 pixels. Este procedimento foi realizado para garantir que os dados de treinamento apresentem sempre a mesma dimensão, permitindo que a mesma arquitetura de rede seja utilizada para treinar todos os membros da população.

Inicialmente, cada canal dos arquivos resultantes, cuja dimensão é Nx1, foi convertido para uma matriz quadrada de ordem $\sqrt{N}$ x $\sqrt{N}$.

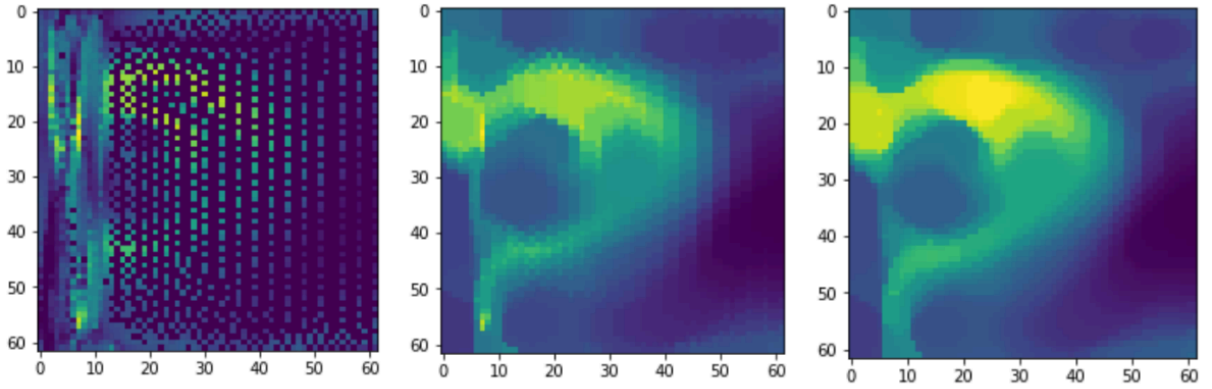
Em sequência, os elementos foram ordenados em relação às coordenadas do elemento em X e Y para que cada elemento ocupasse sua posição correta em relação a imagem quadrada. Cada eixo foi escalonado em números inteiros entre 0 e $\sqrt{N}$. Desta maneira, por exemplo para uma das geometrias estudadas, a superfície do fundo da cavidade foi descrita por 3761 elementos. Após a conversão da imagem, ela foi representada como uma matriz de tamanho 62 x 62 e as coordenadas das colunas das posições X, Y e Z substituídas por números inteiros entre 0 e 62.

Após este processo, os elementos foram ordenados de maneira que ocupassem cada um sua respectiva posição em relação a imagem de tamanho 62 x 62.

Após o escalonamento e ordenação, a imagem resultante apresenta perda de informação, em função do escalonamento de posição possuir apenas números inteiros.

A fim de corrigir esta condição, uma sequência de transformações morfológicas foram utilizadas para preencher os campos não preenchidos pelo processo de ordenamento. Foi utilizado um processo de abertura morfológica, considerando-se um kernel elíptico de tamanho 5x5 com 2 iterações, seguido da aplicação de um filtro de mediana, com kernel quadrado 5x5. As etapas de processamento até a obtenção da imagem final são mostradas na Figura 27.

Figura 27 - Processo de transformação morfológica e filtragem para pré-processamento dos dados de treinamento.

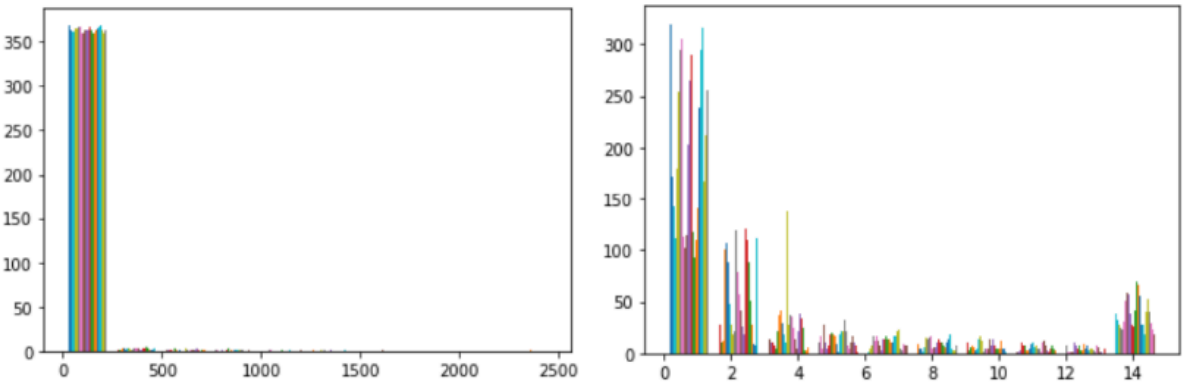


Fonte: Do autor

Posteriormente, a imagem final foi convertida para o tamanho 128 x 128 pixels, para cada membro do treinamento. Todas as transformações e filtragens foram realizadas utilizando-se da biblioteca *OpenCV* (OPENCV, 2021).

Além do processo de transformação morfológica para cada membro da população, foi observado que a simulação gera alguns valores atípicos para os perfis de cisalhamento e fração volumétrica em pixels específicos do domínio. Isto é ocasionado por uma das diversas etapas de geração da malha, execução da simulação ou pré-processamento do arquivo com os resultados da simulação. O histograma mostrado na Figura 28 demonstra a distribuição de valores de cisalhamento por pixel, considerando-se cada pixel dos 369 membros da população de treinamento. Podemos notar que grande parte da população se concentra em faixas de cisalhamento inferiores a 200 Pa, porém existem alguns elementos residuais com valores de cisalhamento muito superiores, na casa de 2500 Pa. Do ponto de vista prático, entende-se que estes elementos em específico não agregam informações relevantes ao modelo, uma vez que os valores mais baixos de cisalhamento, aqueles onde poderia ocorrer resíduo de sabão, são os mais relevantes. Além do mais, uma vez que o treinamento da rede se dá elemento a elemento, estes valores atípicos poderiam aumentar excessivamente o gradiente de treinamento da rede, aumentando a dificuldade de convergência. Desta maneira, aplicou-se uma técnica de *hard-thresholding* (MALLAT, 2009), onde o valor de limitação foi definido no ponto de 90% de ocorrências na função densidade de probabilidade dos dados de treinamento. Valores acima da limitação foram substituídos pelo valor do limite de limitação em si. A Figura 28 demonstra os histogramas para as distribuições de cisalhamento antes e após o processo de *hard-thresholding.*

Figura 28 - Histograma de tensão de cisalhamento antes e após processo de hard-thresholding



Histograma de cisalhamento antes do processo de *hard thresholding*.

Histograma de cisalhamento após o processo de *hard thresholding*.

Fonte: Do autor

Observa-se que após a limiarização, os valores ficaram concentrados entre 0 e 14 Pa.

4.4. Arquitetura de rede neural convolucional

Após o processo de pré-processamento dos dados de treinamento, avançamos para a criação da arquitetura da rede neural. A rede neural investigada neste estudo apresenta como camada de entrada uma quantidade de neurônios igual a 15, considerando-se as 14 variáveis geométricas e a variável de vazão, descritas na seção 4.1. Posterior a esta primeira camada, a rede possui uma camada oculta, cujo objetivo é amplificar a quantidade de neurônios e pesos dos 15 neurônios iniciais. Por último, uma última camada oculta de tamanho 16*16*f x 1 é criada, onde f é o número de filtros convolucionais. Posteriormente, essa camada é convertida para dados estruturados de forma 16 x 16 x f, de maneira que possa ser processada por um filtro de convolução.

Sobre este tensor foi aplicado um módulo de convolução muito similar ao utilizado em Wang et al. (2021) e Du et al. (2021). O módulo de convolução criado consiste em uma camada de conversão de *kernel* tamanho 3, *stride* tamanho 1 e *padding* tamanho 1, além de uma função de ativação *ReLU*.

A definição do *padding* 1 e *stride* 1 são importantes para garantir que após o processo de convolução não ocorra redução do tensor de entrada. Após a aplicação

da convolução e transformada, os dados passaram por um aumento da taxa de amostragem baseado em interpolar cada conjunto de 2 pontos nas duas direções da matriz, de maneira que o tensor final de entrada de tamanho 16 x 16 passe a ter tamanho 32 x 32 (PEDRINI, 2008).

Este processo foi repetido sucessivamente mais duas vezes, de maneira que o tamanho final do domínio após quatro camadas de convolução será de 128 x 128, mesmo tamanho da matriz gerada pelos dados de CFD.

Após as camadas de convolução e expansão, foi realizada mais uma sequência de convoluções com *padding* 1 e *stride* 1 de maneira a manter o tamanho de saída. A quantidade de camadas foi definida experimentalmente.

Após as camadas de convolução, uma última camada foi criada com 128*128x1 neurônios, de maneira que possa ser utilizada para o cálculo de erro em relação aos dados de treinamento com uma função de perda mais simples.

A função de perda atribuída foi o erro absoluto médio, que computa o erro obtido ponto a ponto em campo vetorial real vs. o campo gerado pela rede neural artificial. Esta função de perda foi selecionada por se tratar do cálculo do erro entre grandezas contínuas entre os dados de saída e os dados reais. Além do mais, ela foi preferida em detrimento da função de perda de erro médio quadrático por ter atualização de pesos mais suave e gerar gradientes muito excessivos. Observou-se de maneira empírica que esta função de perda reproduziu melhor os perfis escalares para as duas grandezas avaliadas.

Por último, quanto ao processo de otimização da rede, entende-se que há uma certa combinação de hiperparâmetros que tendem a reduzir o valor da função de perda e melhorar o treinamento da rede. Considerou-se os seguintes hiperparâmetros como mais críticos para o treinamento da rede: Número de camadas convolucionais após o processo de convolução e amplificação, número de neurônios na primeira camada oculta, taxa de *dropout* nas camadas ocultas, tipo de interpolador das camadas ocultas, número de filtros de convolução e tipo de otimizador, entre Adam e descida do gradiente estocástica.

Para o critério de parada da rede neural, os dados de treinamento foram divididos em 70% treinamento, 20% validação e 10% teste. A função *earlyStopping* da biblioteca Keras foi utilizada para que o treinamento fosse interrompido após 20 épocas, caso a função de perda de validação não reduzisse ao menos 0,002 nesta quantidade de épocas.

A fim de avaliar os efeitos dessas seis variáveis independentes sob a rede, realizou-se inicialmente, um experimento do tipo DOE (*Design of experiments)* fracionado, em que o efeito individual e o efeito das interações de segunda ordem dos fatores pôde ser avaliada. Para cada um dos tratamentos, foram realizadas duas réplicas de treinamento, totalizando 32 treinamentos para a rede de cisalhamento e 32 treinamentos para a rede de fração volumétrica. Por último, através da aplicação da análise de ANOVA (*Analysis of Variance*), pode-se identificar as variáveis da rede que de fato tem um efeito significativo na melhora do modelo de treinamento. A Figura 29 abaixo mostra o planejamento experimental realizado, assim como os níveis de cada variável independente neste primeiro experimento (MONTGOMERY, 2017).

Figura 29 - Experimento fatorial fracionado realizado para cada campo vetorial

Simulation setup	1																													
Parametric source model	1																													
Field extraction method	1																													
Product inclination	1																													
Water temperature	1																													
Water flow rate	1																													
X1: FInal convolutional layers	-															+														
X2: Neurons in hidden layer	-				+											-				+										
X3: Dropout hidden ayers	-		+		-		+		-		+		-		+		-		+											
X4: Upsampling interpolators	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+										
X5: Convolution filters	-	-	+	+	+	+	-	-	+	+	-	-	-	-	+	+														
X6: Training optimizer	-	+	+	-	+	-	-	+	-	+	+	-	+	-	-	+														
Replicate	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Y1: Val_loss																																

	-	+
X1	1 layer	3 layers
X2	2048 neurons	4096 neurons
X3	20%	40%
X4	bilinear	bicúbico
X5	16 filtros	32 filtros
X6	Adam	SGD (lr = 0.1)

Fonte: Do autor

Um segundo experimento do tipo DOE, porém completo, foi realizado posteriormente apenas com as três variáveis mais críticas em duas réplicas para cada um dos campos. Após a execução deste segundo experimento, a rede final foi definida como a melhor combinação entre os 6 hiperparâmetros avaliados para cisalhamento e fração volumétrica independentemente. Por fim, para a estimativa dos resultados finais, foi utilizado um comitê de 10 redes neurais. Cada rede neural foi treinada utilizando-se 70% dos dados de treinamento. Esta prática tem

fundamento na literatura, sendo reconhecida por melhorar a acuracidade da estimativa e reduzir o *overfitting* (LECUN, 1998).

4.5. Métricas de avaliação do desempenho do modelo

Esta seção tem como objetivo detalhar as métricas utilizadas para a avaliação de desempenho de treinamento das redes. Conforme mencionado na Seção 4.4, a função de perda utilizada na otimização do modelo foi a função *meanAbsoluteError,* ou MAE. Esta função de erro foi selecionada pelo problema de otimização tratar-se especificamente de um problema contínuo e, além disso, por empiricamente ter se mostrado mais eficiente em reproduzir o perfil observado na simulação.

Conforme detalhado no Capítulo 4.1, o banco de treinamento foi gerado a partir de um experimento do tipo fatorial do tipo DOE, sendo possível obter um modelo de regressão de mínimos quadrados em função de cada termo. Como o erro pode ser calculado para cada membro da população independentemente, foi criado um modelo de regressão da função de perda para cada variável independente do modelo paramétrico. Desta maneira, foi possível entender em quais regiões a confiabilidade da estimativa da rede é melhor e em quais será necessário em trabalhos futuros um aumento da população de treinamento. Esta abordagem é interessante por permitir o uso da ferramenta e estimar o erro esperado para aquela configuração individual em relação a uma simulação computacional executada para aquele mesmo modelo.

5. RESULTADOS

Este capítulo tem como objetivo mostrar os resultados obtidos com o treinamento das redes neurais convolucionais como modelo de substituição para os dois campos escalares de interesse: Cisalhamento e fração volumétrica.

A Seção 5.1 tem como objetivo detalhar os estudos realizados para otimização dos hiperparâmetros de rede, tanto para a rede de cisalhamento quanto para a rede de fração volumétrica, incluindo as análises prática, gráfica e estatística de cada experimento fatorial executado na otimização dos hiperparâmetros.

Na Seção 5.2 serão detalhados aspectos gerais da rede neural otimizada para cada um dos campos escalares e algumas análises qualitativas, comparando perfis de cisalhamento real com aqueles estimados pela rede neural.

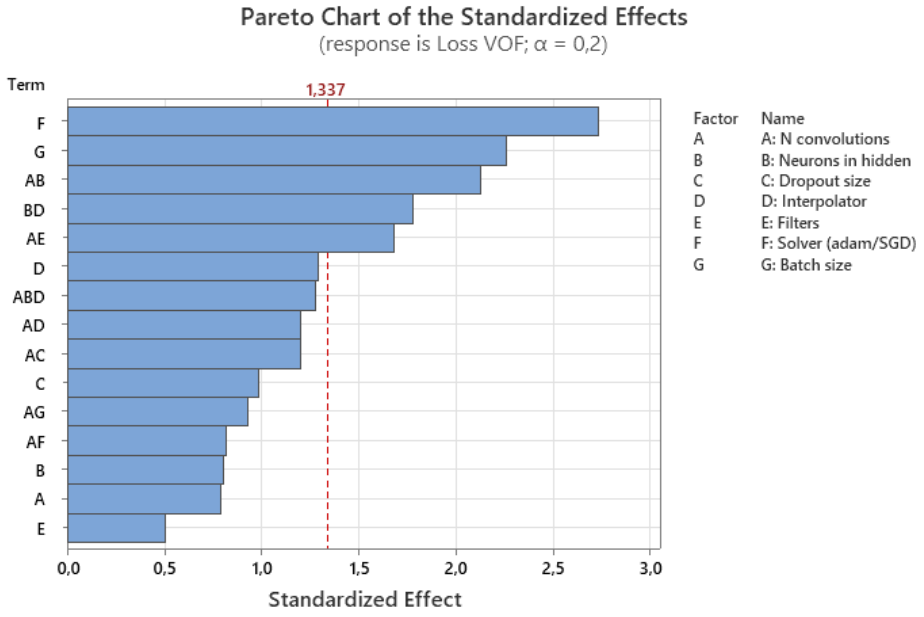
Na Seção 5.3, será analisado o modelo de regressão do erro em função de cada variável independente. Será demonstrado como o modelo de regressão é capaz de estimar o erro da estimativa da rede para cada nível da variável independente do modelo paramétrico, além da vazão. Tanto as análises dos experimentos fatoriais quanto a análise de ANOVA foram executadas utilizando-se o software *Minitab* (MINITAB, 2023).

5.1. Otimização de hiperparâmetros das redes neurais

O estudo de otimização de hiperparâmetros das redes neurais, conforme descrito na Seção 4.4, foi baseado em um estudo fatorial envolvendo 6 variáveis independentes, sendo elas hiperparâmetros da rede.

A Figura 30 abaixo, denominada gráfico de pareto, denota a importância relativa dos efeitos calculados de cada variável independente do modelo em relação ao treinamento, para o campo vetorial de fração volumétrica, ou VOF.

Figura 30 - Gráfico de Pareto do efeito dos graus de liberdade para erro de fração volumétrica



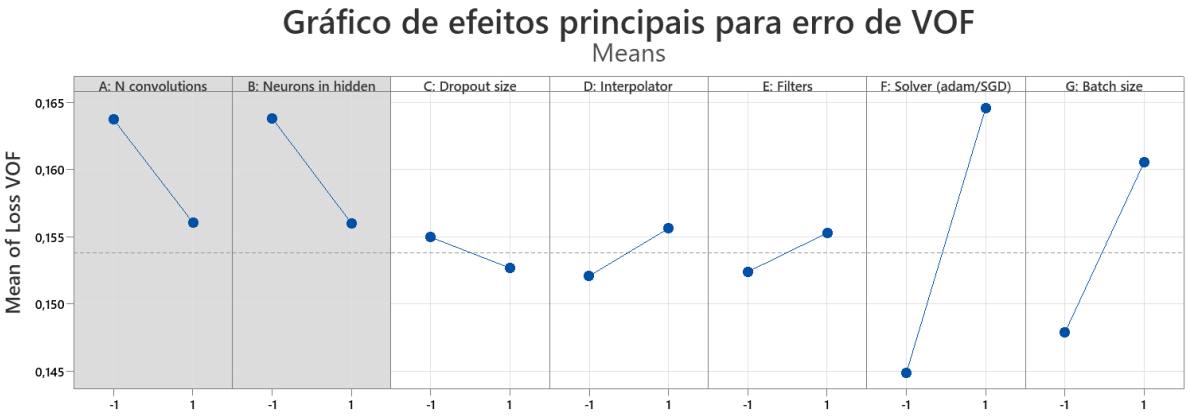
Fonte: Do autor

A linha vermelha neste gráfico refere-se a linha em que a hipótese nula do teste de ANOVA, onde as médias de todas as populações são iguais, ou em outras palavras, que o efeito dos fatores não pode ser diferenciado de variação aleatória do experimento. O termo α=0.2 é relacionado ao risco alfa da significância estatística, ou seja, o grau de certeza que aquele grau de liberdade específico é significante. Desta maneira, as barras no gráfico acima da linha de significância estatística são aqueles fatores que de fato interferem na eficiência do modelo, em um nível diferenciável de mero erro de experimentação, que naturalmente ocorre nas redes neurais, por ser um processo heurístico. Desta maneira, os graus de liberdade significativos podem ser entendidos como os fatores principais F e G, além de interações de dois fatores.

Além dos fatores principais, foram observadas interações significativas. Em função do experimento ter sido executado de maneira fracionada, ocorre o que chamamos de confundimento, onde interações de maior ordem são confundidas com outras interações. Após a quebra destes confundimentos, considerando-se o princípio da hereditariedade, foi possível entender os efeitos dos fatores principais e interações críticas (MONTGOMERY, 2017).

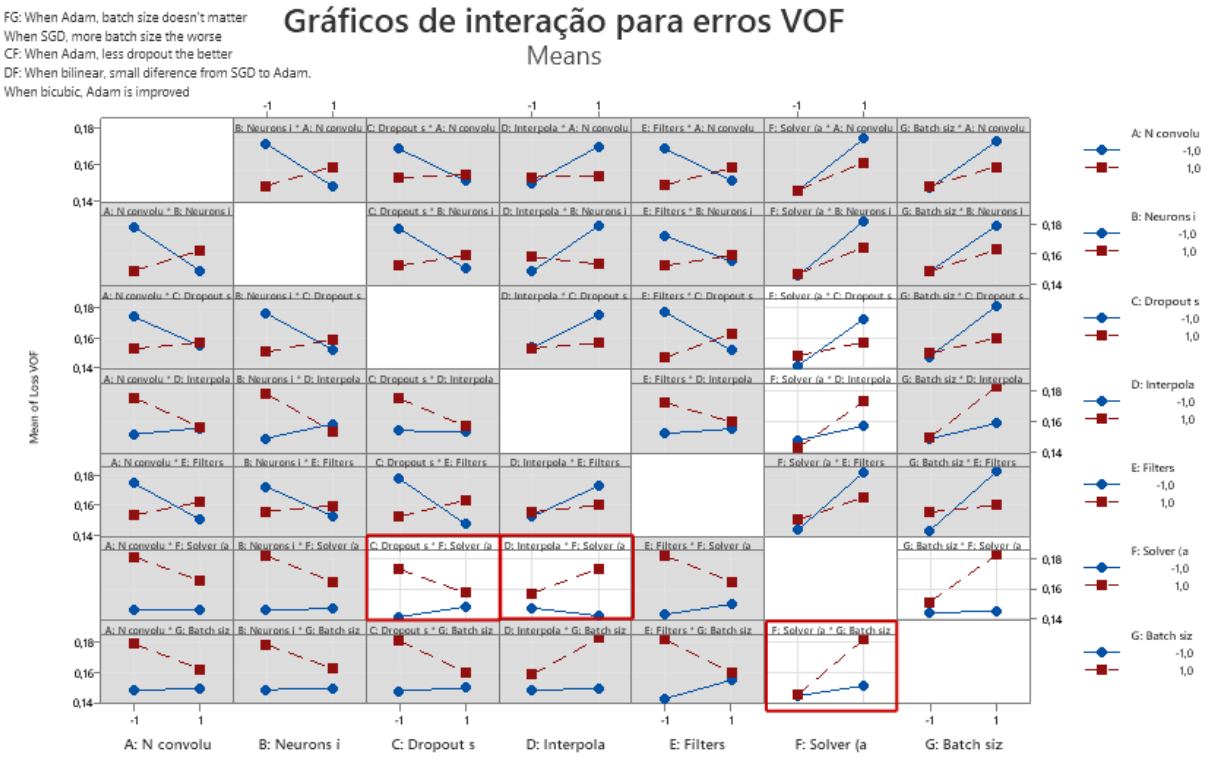
As Figuras 31 e 32 demonstram resultados obtidos com a otimização dos hiperparâmetros da rede de fração volumétrica.

Figura 31 - Gráfico de efeitos principais para erro de VOF



Fonte: Do autor

Figura 32 - Gráfico de interação para erro de VOF



Fonte: Do autor

Como conclusão geral do primeiro experimento, foi possível verificar que: a) O otimizador Adam performou melhor que o otimizador SGD; b) *Batch sizes* menores apresentaram resultados melhores de treinamento; c) Uma quantidade menor de

filtros apresentou melhores resultados; d) Interpolação bilinear para as etapas de aumento amostral entre convoluções foi melhor que bicúbico.

Apesar da interpolação bilinear se mostrar melhor, uma interação significativa mostrou que, quando o otimizador Adam é utilizado, o erro utilizando-se o interpolador bicúbico é reduzido. Demais interações demonstraram que ao se utilizar do otimizador Adam, o efeito de *batch size* é minimizado. Observou-se também que o erro é reduzido, quando se usa o otimizador Adam, ao reduzir-se a porcentagem de *dropout* do sistema.

Um segundo experimento fatorial completo foi executado com um total de 3 fatores e 4 réplicas por fator, conforme mostrado na Figura 33.

Figura 33 - Segundo experimento fatorial para o campo *VOF*

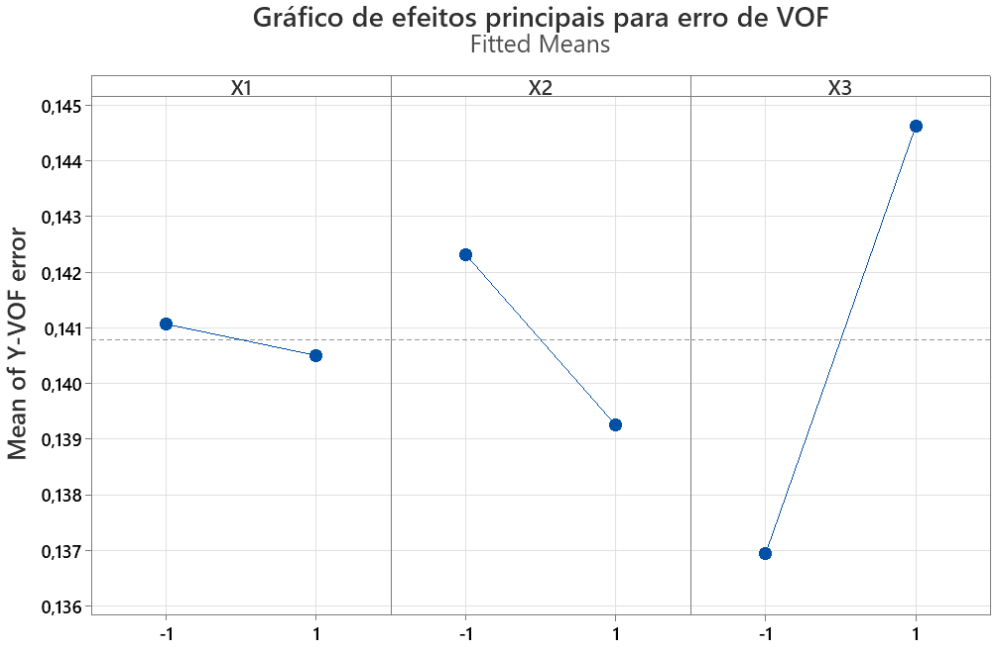
Configuração simulação	1																													
Modelo fonte	1																													
Extração campo	1																													
Inclinação produto	1																													
Temperatura água	1																													
X1: Dropout	-															+														
X2: Filtros	-								+							-								+						
X3: Tamanho Kernel	-		+				-				+				-				+				-				+			
Replicas	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Y1: Val_loss																																

	-	+
X1	10%	20%
X2	8 filtros	16 filtros
X3	3 x 3	5 x 5

Fonte: Do autor

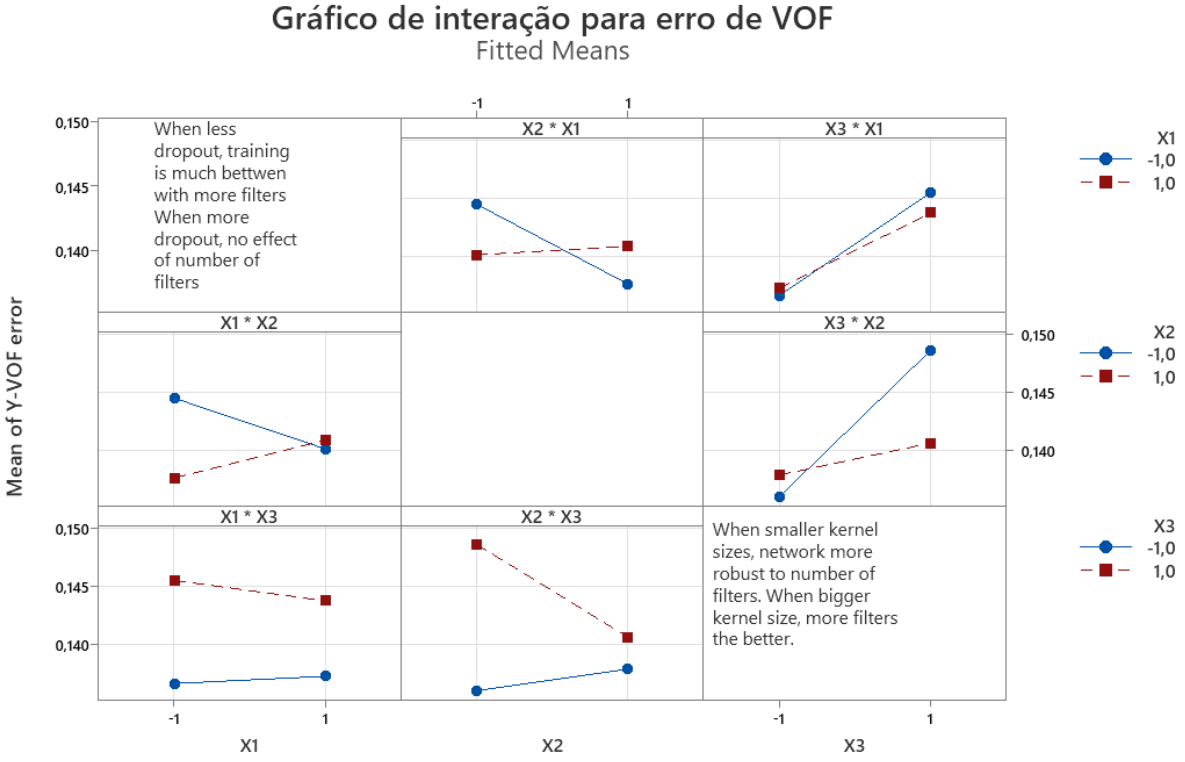
Neste segundo experimento foram mantidos constantes o número de neurônios na camada oculta em 2048, o interpolador como bicúbico, em função de uma interação crítica com o interpolador, o otimizador como Adam e o número de convoluções em 1, após a última camada de amplificação. Foram variados, conforme mostrado acima, o *dropout* entre 10 e 20%, o número de filtros entre 8 e 16 e foi adicionado o tamanho do *kernel* de convolução como fator, entre 3x3 e 5x5 pixels. Os gráficos de efeito principal e de interação são mostrados nas Figuras 34 e 35.

Figura 34 - Gráfico de efeitos principais para segundo experimento fatorial para VOF



Fonte: Do autor

Figura 35 - Gráfico de interações para segundo experimento fatorial para VOF



Como conclusão do segundo experimento, foi possível verificar que: a) pouco efeito de reduzir o *dropout* a menos de 20%; b) Utilizar 16 filtros apresenta melhor performance que 8 filtros; c) O *kernel* de convolução 3x3 apresenta melhores resultados que o *kernel* 5x5; d) Quando a rede possui menos *dropout,* treinamento melhora com mais filtros; e) Com mais *dropout*, independe do número de filtros; f) Quando o *kernel* 3x3 é utilizado, a rede se torna mais robusta ao número de filtros; g) Quando o *kernel* é maior, mais filtros são necessários.

A partir dos dois experimentos executados, o modelo ótimo para treinamento de VOF foi definido com os seguintes parâmetros: 2048 neurônios na camada oculta, 10% de dropout em camadas ocultas, interpolador bicúbico, 16 filtros de convolução, *kernel* de convolução 3x3, otimizador Adam. Esta rede foi treinada por 10 vezes, cada vez fazendo uma nova divisão de treinamento e validação.

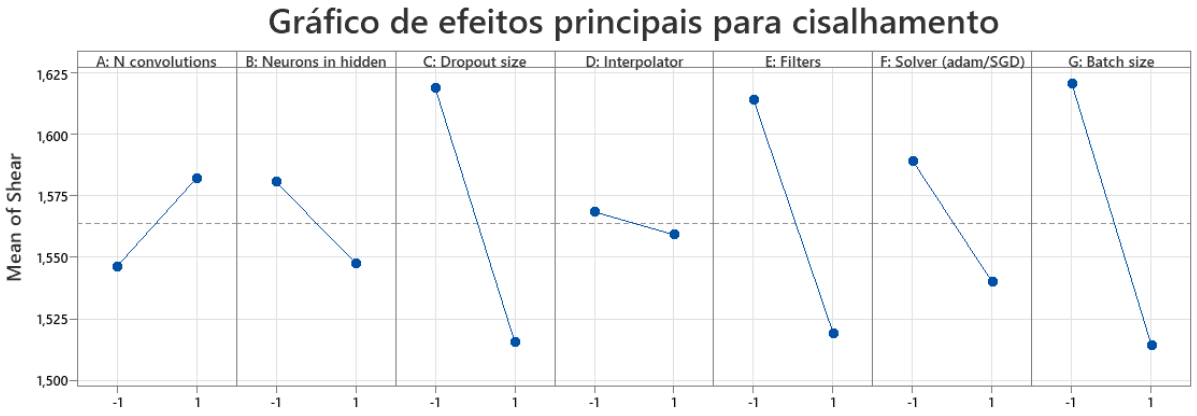
A previsão do campo VOF foi então feita como a média aritmética das 10 redes neurais, para cada pixel individual. O erro médio absoluto para os tratamentos foi de 0,086, muito inferior à média de erro para cada rede neural individual. O desvio padrão foi de 0,0466.

Cada uma das redes neurais do comitê de redes levou cerca de 2 horas para completar o treinamento. Desta maneira, a obtenção do modelo de substituição leva em torno de 20 horas. Uma vez treinado, o tempo de previsão é desprezível. O treinamento foi realizado em um processador *Intel i9-12950HX,* com 32 gb de memória RAM. O computador no qual a simulação foi executada também dispõe de uma GPU NVIDIA RTX A2000 com 8 GB de memória RAM.

Em comparação, caso um simples tratamento fosse submetido para simulação via *CFD*, o tempo de simulação de um tratamento individual levaria entre 24 e 48 horas. Porém, o tempo efetivo para obtenção da resposta, envolvendo priorização da simulação pelo time de simulação, elaboração da malha, execução da simulação e análise dos resultados pode levar até 4 semanas, considerando-se uma alta volatilidade de tempo por ser dependente da demanda para o time de simulação.

Realizou-se o mesmo procedimento, considerando-se o campo vetorial de cisalhamento. Através dos dados experimentais observou-se que o cisalhamento apresentou um conjunto de hiperparâmetros diferente do que o campo de fração volumétrica. A Figura 36 abaixo mostra os efeitos principais dos fatores.

Figura 36 - Gráfico de efeitos principais para o erro de cisalhamento.



Fonte: Do autor

Observou-se que o aumento da fração de *dropout*, o aumento no número de filtros e o aumento do *batch size* reduziram o erro de cisalhamento.

Um segundo experimento foi executado, considerando-se também um estudo fatorial completo. Para este estudo, o otimizador foi mantido constante como *SGD*, o número de convoluções após amplificação fixa em 1, *batch size* fixo em 16. Para este estudo, os fatores estudados foram o número de neurônios na camada oculta, a porcentagem de *dropout* e a quantidade de filtros de convolução. O estudo é ilustrado na Figura 37.

Figura 37 - Segundo experimento fatorial para o Cisalhamento

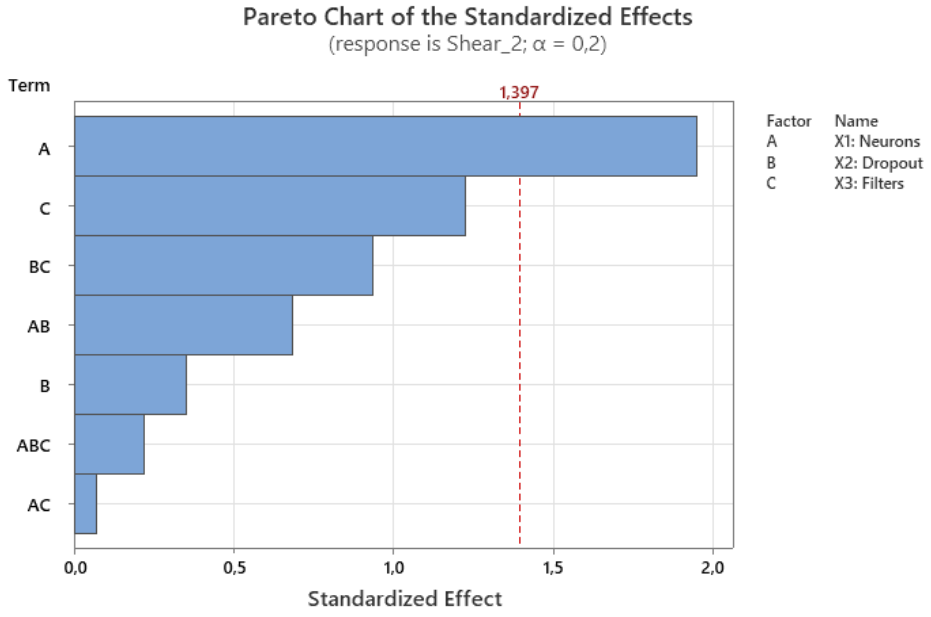
Otimizador	Adam																													
Configuração simulação	1																													
Modelo fonte	1																													
Extração campo	1																													
Inclinação produto	1																													
Temperatura água	1																													
X1: Neurônios camada oculta	-															+														
X2: *Dropout*	-								+							-								+						
X3: Filtros de convolução	-		+		-		+		-		+		-		+		-		+		-		+		-		+			
Replicas	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Y1: Val_loss																																

	-	+
X1	4096	8192
X2	40%	50%
X3	32	128

Fonte: Do autor

Para este experimento, o gráfico de Pareto evidenciou que o fator mais significativo na redução do erro de treinamento foi a quantidade de neurônios na camada oculta, conforme destacado no gráfico da Figura 38, seguido pela quantidade de filtros.

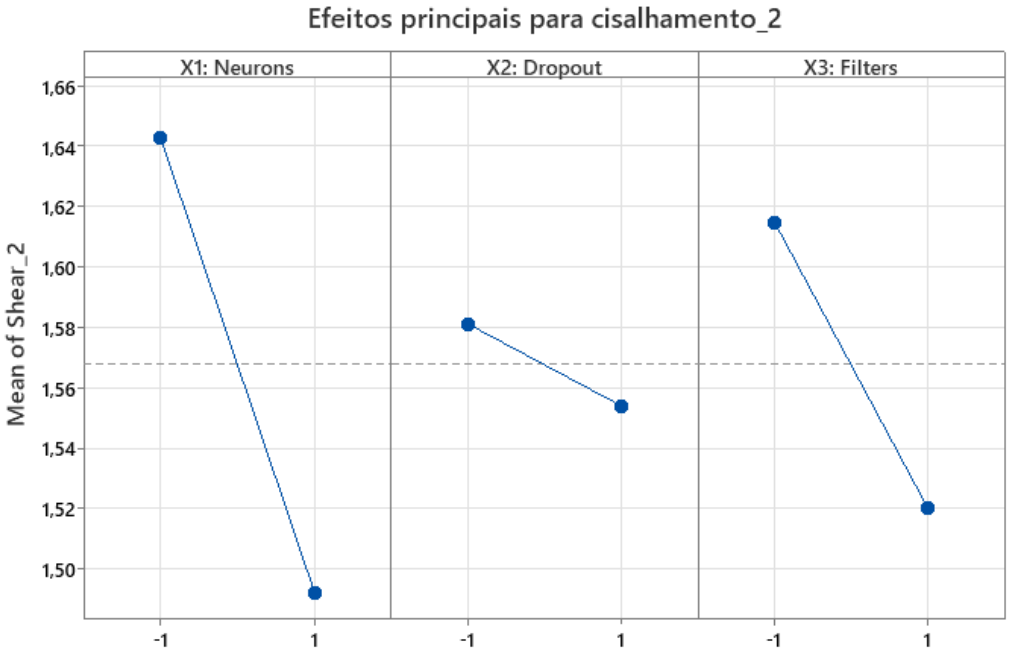
Figura 38 - Gráfico de Pareto do segundo experimento de cisalhamento



Fonte: Do autor

A Figura 39 mostra os efeitos principais para o cisalhamento. Fica claro no gráfico o pouco efeito do aumento de *dropout* no treinamento de cisalhamento, mas os efeitos positivos de se aumentar a quantidade de filtros no treinamento de cisalhamento, apesar do efeito estatístico ficar limítrofe para $\alpha = 0,2$.

Figura 39 - Gráfico de efeitos principais do segundo experimento de cisalhamento



Fonte: Do autor

Novamente, os hiperparâmetros ótimos da rede de cisalhamento foram definidos como segue: 8192 neurônios na camada oculta, 40% de *dropout* em camadas ocultas, interpolador bicúbico, 128 filtros de convolução, *kernel* de convolução 3x3, otimizador descida do gradiente estocástico com taxa de aprendizado de 0,1. Esta rede foi treinada por 10 vezes, cada vez fazendo uma nova divisão de treinamento e validação.

A previsão do campo de cisalhamento foi então feita como a média aritmética das 10 redes neurais, para cada pixel individual. O erro médio absoluto para os tratamentos foi de 0,75, inferior à média de erro para cada rede neural individual. O desvio padrão foi de 0,57. Conforme será demonstrado nas Seções 5.2 e 5.3, diferentemente da rede de VOF que apresentou comportamento extremamente aderente em praticamente todo o espaço de inferência, a rede de cisalhamento apresentou maiores gradientes de erro em função das regiões do espaço de inferência.

As redes de cisalhamento levaram em torno de 12 horas de treinamento para cada uma das redes do comitê, totalizando 120 horas para o treinamento das 10 redes. Foi utilizado o mesmo processador para realização do treinamento que fora utilizado para o treinamento das redes neurais de fração volumétrica.

5.2. Análise qualitativa dos campos escalares

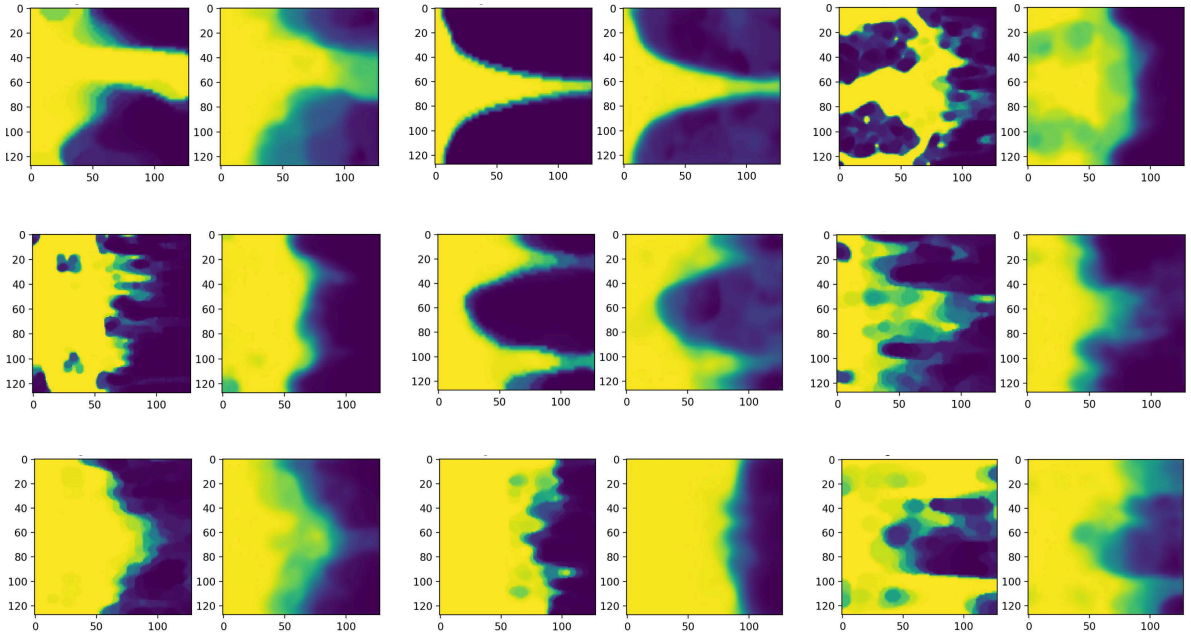
Nesta seção, os campos escalares obtidos via simulação são comparados com os campos escalares gerados pelas simulações de CFD. Conforme mencionado anteriormente, hoje grande parte das conclusões a respeito dos resultados da simulação são analisadas de maneira qualitativa. Além do mais, entender que a análise qualitativa funciona para diferentes perfis significa que a rede aprendeu, de maneira eficaz, a reproduzir diferentes perfis dos campos escalares em variadas condições de entrada, o que infere que de fato houve uma relação entre os pesos aprendidos pela rede e o comportamento físico do sistema.

A Figura 40 abaixo apresenta 9 comparações gráficas entre os resultados gerados pela rede neural em comparação aos resultados obtidos via simulação para a variável resposta VOF. Em cada par de imagens, a imagem da esquerda representa o perfil gerado pelo software de CFD, enquanto a imagem da direita representa o perfil gerado pelo modelo de substituição.

Fica muito claro nas imagens que as estimativas das redes, lembrando que os resultados foram estimados como uma média aritmética de um comitê de 10 redes neurais, foi muito eficaz em reproduzir qualitativamente o comportamento do escoamento do fluido sobre a superfície. Observa-se que entre as imagens abaixo, o perfil de preenchimento com água apresentou comportamentos muito distintos, entre eles: Escoamento predominantemente central, preenchimento apenas parcial do fundo da câmara e fluxo bifurcado. Observa-se que a rede tende a apresentar interfaces menos abruptas entre a região seca (em azul) e as regiões molhadas (em amarelo). Isto se dá especificamente pelo processo de amplificação da rede, considerando-se que a arquitetura possui como entrada 15 variáveis independentes e uma imagem 128x128 pixels como saída. Estes resultados mostram como a técnica tem potencial promissor como ferramenta de desenvolvimento de produto, especialmente por, além de prover resultados quantitativos equiparáveis a

simulação, propicia uma análise de coerência física de acordo com as teorias previamente levantadas pelo engenheiro.

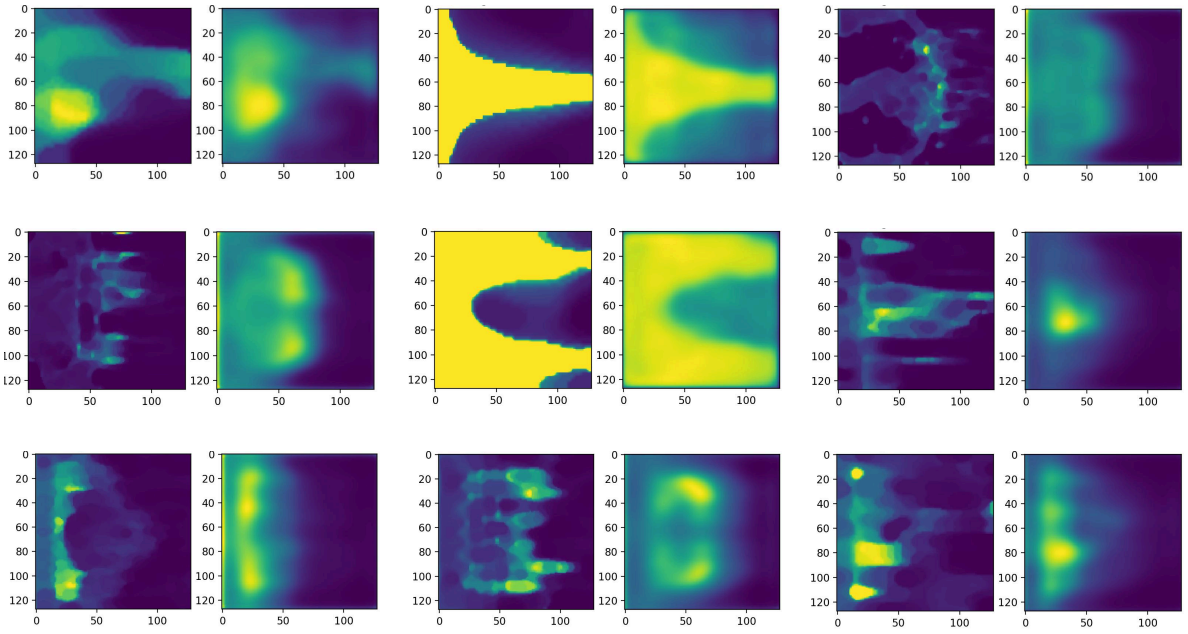
Figura 40 - Comparação de resultados para 9 diferentes perfis de fração volumétrica, comparando resultados de simulação vs. estimativas do comitê de redes neurais



Fonte: Do autor

A mesma análise foi realizada para o campo de cisalhamento (*shear*). Os mesmos tratamentos retratados na Figura 40 são repetidos abaixo, na Figura 41.

Figura 41 - Comparação de resultados para 9 diferentes perfis de cisalhamento, comparando resultados de simulação vs. estimativas do comitê de redes neurais



Fonte: Do autor

Diferentemente do campo de fração volumétrica, observa-se que o campo de cisalhamento apresenta regiões com gradientes bem maiores. Isto é uma característica esperada na aplicação, considerando-se regiões onde há picos de cisalhamento as regiões onde predominantemente o fluxo de água proveniente do espalhador atinge a cavidade com maior velocidade. Pode-se observar que o modelo de substituição teve dificuldade em representar esses gradientes elevados, sendo que em perfis onde há uma concentração local de cisalhamento forte, o modelo falhou na previsão. Contudo, observa-se que, em geral, o modelo obteve uma eficácia significativa em prever a região onde o pico de cisalhamento ocorreu e sua magnitude. Desta maneira, ainda que não consiga representar eficazmente os fortes gradientes de tensão de cisalhamento, ainda pode trazer percepções

relevantes sobre o perfil de cisalhamento do componente. Observa-se o mesmo efeito de suavização das bordas, assim como foi observado para os perfis de fração volumétrica gerados pelo modelo de substituição.

Vale destacar também que, baseado na experiência de aplicações anteriores deste tipo de simulação para componentes desenvolvidos de função similar, valores em torno de 3 Pa seriam suficientes para, localmente, evitar a presença de resíduo. Em estudos futuros, as redes de cisalhamento poderiam ser retreinadas, considerando-se uma limiarização mais agressiva dos dados de entrada, limitando-os ao valor de cisalhamento mencionado acima.

Considerando-se que o processo de limiarização limitou valores a até 14 Pa, uma limiarização mais agressiva, limitando o pico de cisalhamento a um valor que empiricamente não gera resíduo de sabão, poderia reduzir a assimetria do histograma de cisalhamento das amostras e, consequentemente, aprimorar os resultados de treinamento da rede.

5.3. Modelo de regressão do erro vs. variáveis de entrada

Esta seção tem como objetivo detalhar e analisar o efeito de cada variável independente em relação ao erro obtido pela rede neural. Esta análise é possível pois, conforme mencionado no Capítulo 4.1, os dados de treinamento foram gerados utilizando-se um experimento fatorial, o que permite a obtenção de um modelo de regressão onde a variável dependente é o erro do treinamento da rede e as variáveis independentes são as variáveis de treinamento. A análise é realizada aplicando-se a técnica de ANOVA (*Analysis of Variance).* A técnica permite comparar o efeito de cada variável independente do modelo e comparar com o erro puro de treinamento, permitindo identificar as variáveis que de fato impactam no aumento ou diminuição do erro.

A análise é baseada em comparar a variância gerada pela mudança de nível de cada fator em relação ao erro puro da regressão. Desta maneira, foram considerados como fatores significativos todos aqueles em que o risco alfa da hipótese nula é inferior a 10%. A Tabela 4 demonstra a análise de ANOVA executada para o campo de fração volumétrica. Podemos verificar que 38 graus de liberdade do modelo se mostraram significativos, considerando-se um erro alfa de 10%. O

coeficiente de determinação (R^2) deste modelo contendo os 38 graus de liberdade foi de 63,77%.

Tabela 4 - Tabela de análise de variância (*Anova*) para a variável VOF

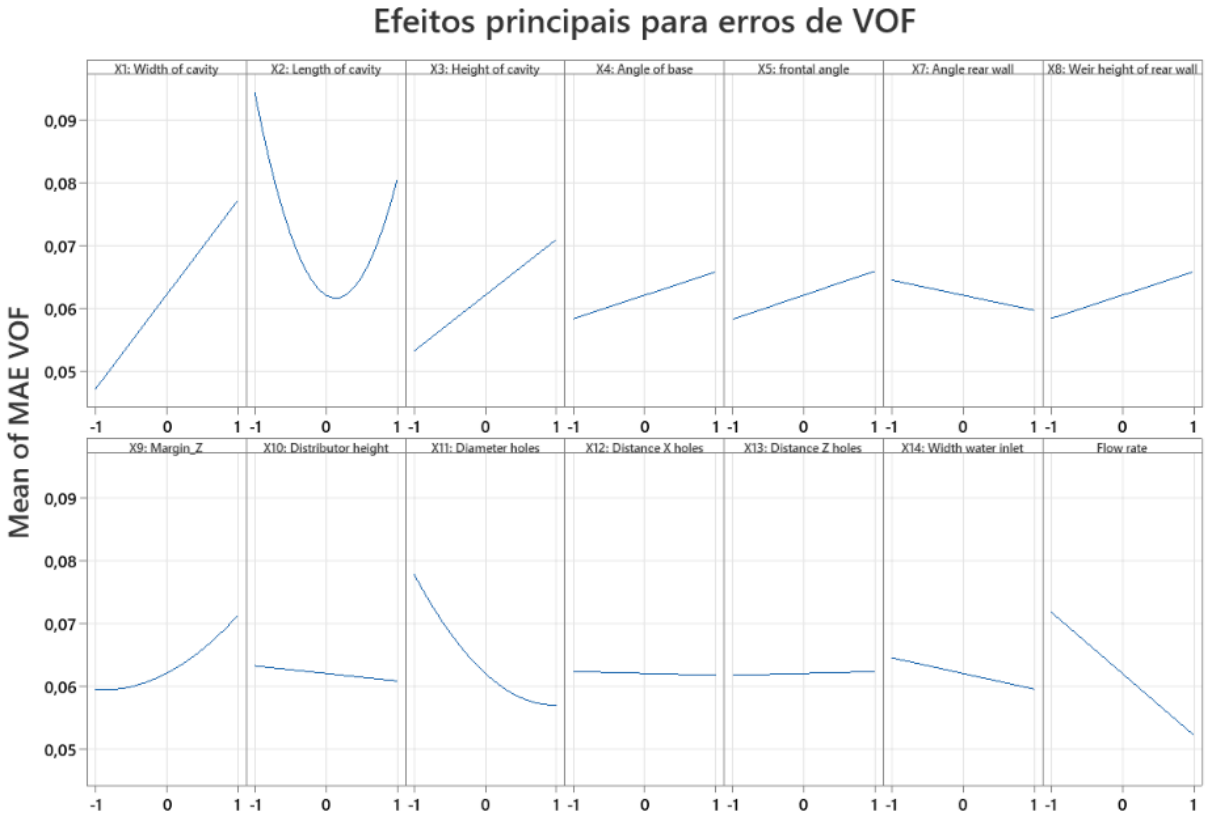
Variável	GL	SS	MS	F-value	P-value
Regressão	38	0,386135	0,010161	11,12	0
X1: Largura cavidade	1	0,044388	0,044388	48,56	0
X2: Comprimento cavidade	1	0,009687	0,009687	10,6	0,001
X3: Altura cavidade	1	0,015545	0,015545	17,01	0
X4: Ângulo da base	1	0,002724	0,002724	2,98	0,086
X5: Ângulo frontal	1	0,002865	0,002865	3,13	0,078
X7: Ângulo parede traseira	1	0,001131	0,001131	1,24	0,267
X8: Altura de transbordo da par	1	0,002714	0,002714	2,97	0,086
X9: Margem Z	1	0,006864	0,006864	7,51	0,007
X10: Altura espalhador	1	0,000285	0,000285	0,31	0,577
X11: Diâmetro dos furos	1	0,021461	0,021461	23,48	0
X12: Distância X furos	1	0,000018	0,000018	0,02	0,887
X13: Distância Z furos	1	0,000017	0,000017	0,02	0,892
X14: Largura entrada de água	1	0,001235	0,001235	1,35	0,246
Vazão	1	0,01782	0,01782	19,5	0
X2: Comprimento cavidade*X2: Comprimento cavidade	1	0,062629	0,062629	68,52	0
X1: Largura cavidade*X4: Ângulo da base	1	0,005757	0,005757	6,3	0,013
X1: Largura cavidade*X5: Ângulo frontal	1	0,016286	0,016286	17,82	0
X1: Largura cavidade*X8: Altura de transbordo	1	0,009835	0,009835	10,76	0,001
X1: Largura cavidade*X9: Margem Z	1	0,019647	0,019647	21,49	0
X1: Largura cavidade*X11: Diâmetro dos furos	1	0,020718	0,020718	22,67	0
X1: Largura cavidade*X14: Largura entrada de água	1	0,005409	0,005409	5,92	0,016
X1: Largura cavidade*Vazão	1	0,007741	0,007741	8,47	0,004
X2: Comprimento cavidade*X11: Diâmetro dos furos	1	0,007228	0,007228	7,91	0,005
X2: Comprimento cavidade*X12: Distância X furos	1	0,005056	0,005056	5,53	0,019
X2: Comprimento cavidade*Vazão	1	0,003994	0,003994	4,37	0,038

X3: Altura cavidade*X4: Ângulo da base	1	0,007883	0,007883	8,62	0,004
X3: Altura cavidade*X5: Ângulo frontal	1	0,007908	0,007908	8,65	0,004
X3: Altura cavidade*X8: Altura de transbordo	1	0,002806	0,002806	3,07	0,081
X3: Altura cavidade*X12: Distância X furos	1	0,010832	0,010832	11,85	0,001
X3: Altura cavidade*X13: Distância Z furos	1	0,014562	0,014562	15,93	0
X4: Ângulo da base*X5: Ângulo frontal	1	0,003065	0,003065	3,35	0,068
X4: Ângulo da base*X14: Largura entrada de água	1	0,003134	0,003134	3,43	0,065
X7: Ângulo parede traseira*X13: Distância Z furos	1	0,004083	0,004083	4,47	0,036
X8: Altura de transbordo da par*X10: Altura espalhador	1	0,013795	0,013795	15,09	0
X9: Margem Z*X 14: Largura entrada de água	1	0,002563	0,002563	2,8	0,095
X11: Diâmetro dos furos*Vazão	1	0,01886	0,01886	20,63	0
X13: Distancia Z furos*Vazão	1	0,002563	0,002563	2,8	0,095
X14: Largura entrada de água*Vazão	1	0,003025	0,003025	3,31	0,07
Erro	240	0,219372	0,000914		
Total	278	0,605507			

Fonte: Do autor

Na Tabela 4 acima, DF representa os graus de liberdade do modelo regressor, SS a variância dos efeitos dos regressores, MS representa a variância ponderada pelos graus de liberdade, *“F-value”* representa a razão entre a variância gerada pelo grau de liberdade e o erro aleatório e *‘P-value’* representa a probabilidade que determinado efeito não é significativo. Quanto menor o valor de ‘P-value’, maior a probabilidade de que o efeito gera uma variância distinguível de uma variância aleatória. O gráfico de efeitos principais ilustrado na Figura 42 por sua vez permite analisar qual o efeito individual de cada grau de liberdade no erro médio do tratamento, em função das configurações de cada variável independente.

Figura 42 - Gráficos de efeitos principais para erros de VOF



Fonte: Do autor

Esta análise de erro por variável é muito importante para futuras iniciativas visando a melhora do modelo e na tomada de decisão do engenheiro utilizando a ferramenta em uma etapa de desenvolvimento de produto. Ela permite, além de uma estimativa pontual do campo escalar gerado pela simulação, do erro em relação a simulação que o modelo de substituição está gerando. Podemos observar, por exemplo, no primeiro quadrante, como o aumento de X1 aumenta o erro estimado do modelo de 0,04 para 0,078. Observa-se também, por exemplo, que o erro de estimativa é reduzido de 0,072 para 0,052, quando a estimativa é feita para valores de vazão máxima em relação à vazão mínima.

A mesma análise de erro foi executada para o campo de cisalhamento. Para o cisalhamento, 16 graus de liberdades se mostraram significantes estatisticamente, com coeficiente de determinação (R^2) de 74%. Um coeficiente de determinação mais elevado para o erro de cisalhamento em relação à fração volumétrica significa que uma proporção maior da variância total do experimento é explicada por erro aleatório do treinamento e 74% é explicada pelos 16 graus de liberdade significativos. A

tabela de ANOVA com os graus de liberdade significativos é ilustrada abaixo, na Tabela 5.

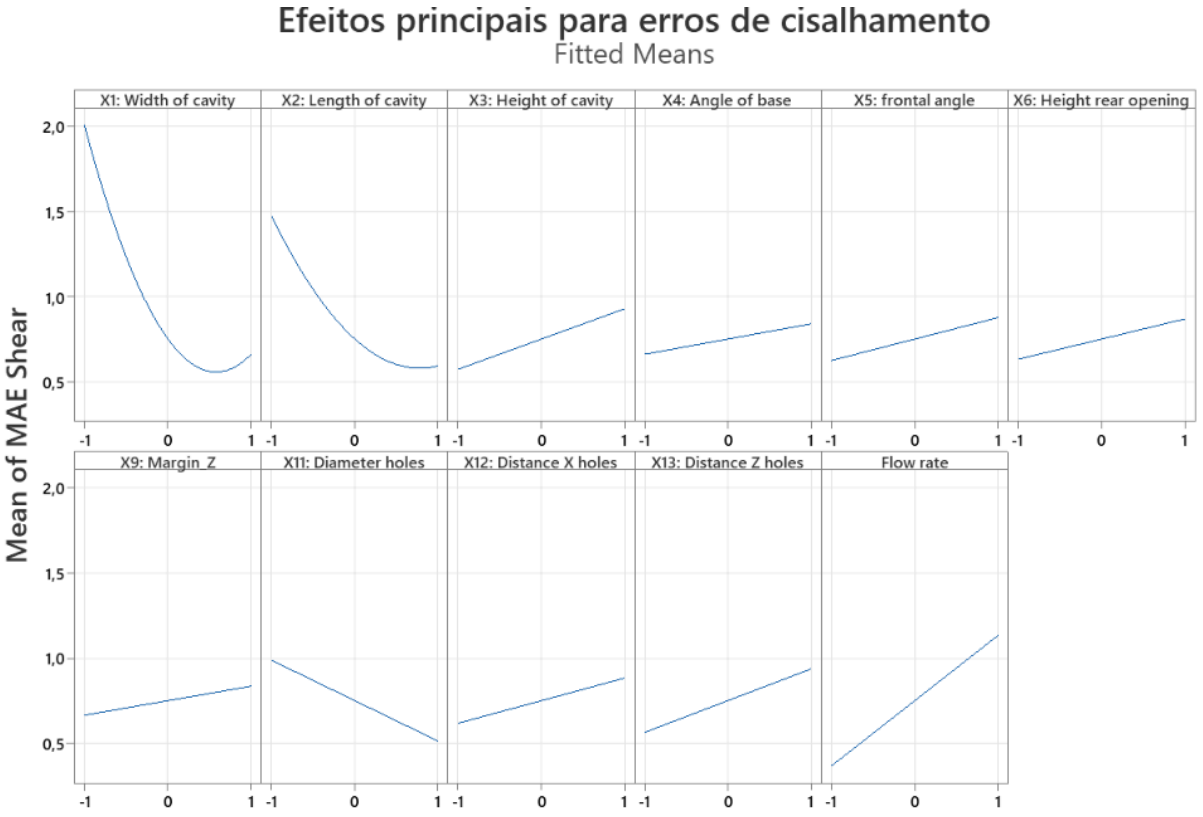
Tabela 5 - Tabela de análise de variância (*Anova*) para a variável Cisalhamento

Variável	GL	SS	MS	F-value	P-value
Regressão	16	245,967	15,3729	47,38	0
X1: Largura cavidade	1	89,301	89,3008	275,2	0
X2: Comprimento cavidade	1	37,923	37,9228	116,87	0
X3: Altura cavidade	1	6,194	6,1939	19,09	0
X4: Ângulo da base	1	1,567	1,5675	4,83	0,029
X5: Ângulo frontal	1	3,141	3,1407	9,68	0,002
X6: Altura abertura traseira	1	2,747	2,7474	8,47	0,004
X9: Margem Z	1	1,448	1,448	4,46	0,036
X11: Diâmetro dos furos	1	11,129	11,1291	34,3	0
X12: Distância X furos	1	3,483	3,4834	10,73	0,001
X13: Distância Z furos	1	6,87	6,8696	21,17	0
Vazão	1	27,474	27,4742	84,67	0
X1: Largura cavidade*X1: Largura cavidade	1	3,872	3,8718	11,93	0,001
X2: Comprimento cavidade*X2: Comprimento cavidade	1	0,896	0,8964	2,76	0,098
X1: Largura cavidade*X6: Altura abertura traseira	1	1,016	1,0164	3,13	0,078
X1: Largura cavidade*X11: Diâmetro dos furos	1	9,069	9,0689	27,95	0
X3: Altura cavidade*X9: Margem Z	1	3,639	3,6389	11,21	0,001
Erro	262	85,017	0,3245		
Total	278	330,984			

Fonte: Do autor

O gráfico de efeitos principais também pode ser analisado para o campo de cisalhamento, conforme mostrado na Figura 43.

Figura 43 - Efeitos principais para erros de cisalhamento



Fonte: Do autor

É possível verificar no gráfico, por exemplo, que o erro de previsão é reduzido consideravelmente ao trabalhar com domínios com X1 maior, assim como X2. Pode-se observar também, em linha com a teoria levantada na Seção 5.2 a respeito da dificuldade do modelo de substituição em representar gradientes abruptos, que o aumento de vazão desloca o erro médio de previsão de 0,4 para em torno de 1,2.

O grande valor desta análise, conforme mencionado anteriormente, é obter um modelo de regressão que forneça ao engenheiro uma estimativa da magnitude do erro de previsão que se está correndo ao gerar os campos com o modelo de substituição. Dependendo da flexibilidade de seleção de variáveis que o engenheiro dispõe, pode optar-se por trabalhar em faixas em que o erro é minimizado. Além do mais, esta análise é complementar à análise da média do erro, sendo que a média do erro pode estar sendo inflada por erros no espaço de inferência em regiões em que é pouco provável que o engenheiro opte por trabalhar. Desta maneira, o erro da estimativa será bem menor que a média.

Caso a região com erro alto não seja uma região com probabilidade baixa de estar em uma zona viável de trabalho, a análise do erro fornece informações a respeito de regiões onde uma maior quantidade de amostras poderia ser gerada, visando melhorar a previsão especificamente nas regiões de erro elevado, através do retreino da rede.

6. CONCLUSÕES

A análise bibliográfica realizada evidencia os benefícios que o uso de ferramentas de aprendizado profundo podem trazer quando relacionadas ao campo de engenharia, especificamente em aplicações envolvendo simulações numéricas e desenvolvimento de produtos.

Ainda que as simulações de fluidodinâmica computacional tragam diversos benefícios nas etapas iniciais de concepção de produto no sentido de reduzir prototipagem e iterações de projeto, devido à complexidade do domínio matemático que envolve a interação fluido-estrutura, estes modelos ainda são por vezes impeditivos de serem aplicados juntamente com algoritmos de otimização, sejam eles heurísticos ou baseados em gradiente, devido ao longo tempo de simulação que cada tratamento único demanda.

Desta maneira, esforços em aprendizado profundo foram focados em desenvolver modelos, denominados modelos de substituição ou *surrogate models*, especialmente baseados em redes neurais, com a capacidade de utilizar dos resultados de simulações previamente executadas para aprender a correlação em relação aos dados de entrada e às variáveis de performance relacionadas ao escoamento.

Posteriormente, dada a limitação destes modelos em representar detalhes específicos da geometria e a falta de capacidade de interpretação dos resultados, foram desenvolvidas redes neurais, principalmente baseadas em arquiteturas convolucionais, que além de aprender apenas as variáveis de performance, aprendem de fato o comportamento do escoamento e os campos de pressão do fluido. Além de melhorar a capacidade de previsão destes modelos, a rede torna capaz a interpretação dos resultados de maneira bem mais direta.

É exatamente neste âmbito que esta dissertação foi desenvolvida. Foi proposta, dentro do escopo deste projeto, uma metodologia de arquitetura de rede neural com o objetivo de servir de modelo de substituição para um tipo específico de aplicação de simulação fluidodinâmica, o dispensamento de sabão em pó por um dispensador automático, utilizado em lavadoras de roupas. O principal diferencial da proposta de rede neural apresentada nesta pesquisa foi utilizar variáveis relacionadas à geometria e condições de uso do sistema como dados de entrada e utilizar como dados de treinamento campos escalares de fato, exportados a partir de

um modelo paramétrico baseado nestas variáveis independentes, executado em um software de fluidodinâmica computacional. A variável de treinamento utilizada foi um campo estruturado, similar a uma imagem digital, contendo para cada pixel os valores de duas grandezas físicas, cisalhamento e fração volumétrica de água, da superfície mais crítica para o processo de dispensamento.

O estudo mostrou que a rede neural foi capaz de aprender a reproduzir os resultados destes campos projetados sobre a superfície de interesse, apresentando resultados qualitativos muito aderentes com os resultados da simulação para uma gama muito variada de comportamentos.

Outro ponto a destacar foi o uso de técnicas de estudos fatoriais do tipo planejamento de experimentos (DOEs), tanto na etapa de geração do banco de dados de treinamento quanto no processo de otimização dos hiperparâmetros da rede, que em si é um processo predominantemente empírico.

A aplicação destas ferramentas permite, para o segundo uso, encontrar uma combinação de hiperparâmetros que apresentou um erro de estimativa muito menor que no início do estudo, otimizando os resultados obtidos.

Para o primeiro uso, na geração do banco de dados, a aplicação permite uma análise de regressão, relacionando as variáveis independentes do modelo com o erro gerado pela rede neural em relação ao resultado da simulação. A grande vantagem desta análise é entender de maneira clara em quais regiões do espaço de inferência das variáveis independentes a reprodução da rede neural foi mais eficaz e em quais iniciativas futuras uma amostragem maior deveria ser utilizada.

Por último, um ponto a destacar é a aplicabilidade da metodologia proposta na Seção 4.3 a qualquer outro tipo de aplicação que envolva uma simulação de fluidodinâmica computacional. As etapas de pré-processamento da imagem proveniente do software de simulação, assim como a elaboração da arquitetura da rede neural e o processo de otimização de hiperparâmetros proposta no Capítulo 4, com respectivos resultados no Capítulo 5, são extensíveis a aplicações aerodinâmicas, de transferência de calor, fluidodinâmica, etc. A ferramenta também não está vinculada a uma grandeza física específica, sendo que a definição das grandezas cisalhamento e fração volumétrica neste estudo em específico se deu por sua aplicabilidade no sistema de estudo.

Por fim, a ferramenta ainda possui diversas oportunidades de melhoria, especialmente nos pontos que tangem o pré-processamento de dados. Conforme

destacado na Seção 4.3, o processo de simulação em si, como o próprio pós processamento e filtragem de imagens pode gerar certos *outliers* que comprometem o treinamento, especificamente pois as funções de perda são relacionadas a variáveis contínuas.

Outra área de melhoria tange a arquitetura de rede neural utilizada no experimento. Em treinamentos futuros, é pertinente avaliar a utilização de redes neurais fisicamente informadas (*physical informed neural networks)* e redes neurais de Kolmogorov, que possuem eficácia em problemas similares, tratando-se do treinamento relacionado a fenômenos físicos (LIU, 2024).

Além disso, esforços podem ser aplicados em entender outros tipos de funções de perda além de erro médio quadrático e erro médio absoluto e seus efeitos no erro de treinamento da rede. Pode-se também entender o efeito do uso de outros tipos de otimizadores, tais como o *RMSprop*, no treinamento (BUSHAEV, 2018).

Por último, foi utilizada a técnica de comitês de redes neurais, o que mostrou-se promissor e efetivo na melhora das estimativas dos campos escalares. Um estudo de refino em relação à quantidade de redes neurais pertencentes ao comitê pode ser realizado, com o objetivo de definir o ponto de inflexão, onde o aumento na quantidade de redes treinadas não gera melhoria no treinamento do modelo.

REFERÊNCIAS BIBLIOGRÁFICAS

ABBAS, Asad et al. **Geometric convolutional neural networks – a journey to surrogate modeling of maritime CFD**. In: Proceedings of the 9th Conference on Computational Methods in Marine Engineering. Edimburgo, 2022. p. 293.

ABBOTT, Ira H.; VON DOENHOFF, Albert E. **Theory of wing sections: including a summary of airfoil data**. Courier Corporation, 2012.

ALVES FILHO, Avelino. **Elementos finitos – a base da tecnologia CAE**. Saraiva Educação S.A., 2018.

AN, Yifu; DU, Xiaosong; MARTINS, Joaquim R. A convolutional neural network model based on multiscale structural similarity for the prediction of flow fields. In: **AIAA Aviation 2021 Forum**. 2021. p. 3061.

ANSYS, Inc. **ANSYS Fluent: Computational Fluid Dynamics Software**. Versão 2024 R1. Canonsburg: ANSYS, Inc., 2024. Disponível em: https://www.ansys.com/products/fluids/ansys-fluent.

ANSYS, Inc. **SpaceClaim: 3D Modeling Software**. Versão 2024 R1. Canonsburg: ANSYS, Inc., 2024. Disponível em: https://www.ansys.com/products/3d-design/ansys-spaceclaim.

DE BOOR, Carl. **A practical guide to splines**. New York: Springer-Verlag, 1978.

BOUHLEL, Mohamed Amine; HE, Sicheng; MARTINS, Joaquim RRA. Scalable gradient–enhanced artificial neural networks for airfoil shape design in the subsonic and transonic regimes. ***Structural and Multidisciplinary Optimization***, v. 61, n. 4, p. 1363-1376, 2020.

BUSHAEV, Vitaly. Understanding RMSprop-faster neural network learning**. *Towards Data Science***, 2018.

BOX, George E.P.; WILSON, K.B. **Central composite designs**. In: MONTGOMERY, Douglas C. *Design and analysis of experiments*. 8. ed. New York: John Wiley & Sons,. p. 305-310,2017.

CAUCHY, Augustin et al. Méthode générale pour la résolution des systèmes d'équations simultanées. ***Comptes Rendus de l'Académie des Sciences***, Paris, n.25: p. 536-538,1847.

CHANG, Angel X. et al. **Shapenet: an information-rich 3D model repository**. *arXiv preprint arXiv:1512.03012*, 2015.

ANSYS Inc. **CFX-solver**. Versão 2024. Canonsburg, PA: ANSYS Inc., 2024. Disponível em: https://www.ansys.com/products/fluids/ansys-cfx.

DU, Xiaosong; HE, Ping; MARTINS, Joaquim RRA. Rapid airfoil design optimization via neural networks-based parameterization and surrogate modeling. ***Aerospace Science and Technology***, n.113: 106701,2021.

DU, Qiuwan et al. Aerodynamic design and optimization of blade end wall profile of turbomachinery based on series convolutional neural network. ***Energy***, n.244: 122617,2022.

ECHEVERRÍA, D.; HEMKER, P. W. **Manifold mapping: a two-level optimization technique**. ***Computing and Visualization in Science***, v.11.n. 4, p.193-206, 2008.

FEY, Matthias et al. Spline Cnn: Fast geometric deep learning with continuous b-spline kernels. **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, p. 869-877,2018.

FIESLER, Emile; BEALE, Russell (ed.). **Handbook of neural computation**. CRC Press, 2020.

GILL, Philip E.; MURRAY, Walter; SAUNDERS, Michael A. SNOPT: An SQP algorithm for large-scale constrained optimization. **SIAM Review**, v.47.n.1: p.99-131,2005.

GOODFELLOW, Ian, et al. Generative adversarial networks. **Communications of the ACM**, v. 63.n.11: p.139-144, 2020.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. MIT Press, 2016.

GRAVES, Alex. Long short-term memory. **Supervised sequence labeling with recurrent neural networks**, Springer, 2012.

HARRIES, S. Practical Shape Optimization Using CFD: State-Of-The-Art in Industry and Selected Trends. **Proceedings of the Conference on Computer Applications and Information Technology in the Maritime Industries**,.p. 17-19, 2020..

HIRT, Cyril W.; NICHOLS, Billy D. Volume of fluid (VOF) method for the dynamics of free boundaries. **Journal of Computational Physics**, v39,n.1,p. 201-225,1981.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. **Neural Computation,**, v.9,n..8,p.1735-1780,1997.

HUNTER, Karen. ScienceDirect™. **The Serials Librarian**, v. 33, n. 3-4, p. 287-297,1998.

JADERBERG, Max, et al. Spatial transformer networks. **Advances in Neural Information Processing Systems**, p.28, 2015.

JASAK, Hrvoje et al. OpenFOAM: A C++ library for complex physics simulations. **International workshop on coupled methods in numerical dynamics**. **IUC Dubrovnik**, Croácia, p. 1-20, 2007.

KINGMA, Diederik P.; BA, Jimmy. Adam: **A method for stochastic optimization**. arXiv preprint arXiv:1412.6980, 2014.

KIRKPATRICK, Scott; GELATT JR, C. Daniel; VECCHI, Mario P. Optimization by simulated annealing. **Science**, v.220, n.4598, p.671-680,1983.

KOZIEL, Slawomir; LEIFSSON, Leifur. **Surrogate-based modeling and optimization.** New York: Springer, 2013.

KRIGE, Daniel G. A statistical approach to some basic mine valuation problems on the Witwatersrand. **Journal of the Southern African Institute of Mining and Metallurgy**, v.52, n.6, p. 119-139,1951.

KUNCHEVA, Ludmila I. **Combining pattern classifiers: methods and algorithms**. John Wiley & Sons, 2014.

LALONDE, Eric Rowland, et al. Comparison of neural network types and architectures for generating a surrogate aerodynamic wind turbine blade model. **Journal of Wind Engineering and Industrial Aerodynamics**, v.216 n.104696,2021.

LECUN, Yann, et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE 86**, n.11, p. 2278-2324,1998.

LI, David X. On default correlation: A copula function approach. **The Journal of Fixed Income,** v.9, n.4, p.43-54,2000.

LI, Jichao; BOUHLEL, Mohamed Amine; MARTINS, Joaquim RRA. Data-based approach for fast airfoil analysis and optimization. **AIAA Journal**, v.57, n.2, p.581-596, 2019.

LIU, Weiyu; BATILL, S. Gradient-enhanced neural network response surface approximations. **8th Symposium on Multidisciplinary Analysis and Optimization.**. p.4923, 2000.

LIU, Ziming, et al. Kan: Kolmogorov-arnold networks. arXiv preprint arXiv:2404.19756, 2024.

LOH, Wei-Liem. On Latin hypercube sampling. **The annals of statistics**, v.24, n.5, p. 2058-2080,1996.

MA, Pingchuan; DU, Tao; MATUSIK, Wojciech. Efficient continuous pareto exploration in multi-task learning. **International Conference on Machine Learning. PMLR**, p. 6522-6531,2020.

MADER, Charles A., et al. ADflow: An open-source computational fluid dynamics solver for aerodynamic and multidisciplinary optimization. **Journal of Aerospace Information Systems**,, v.17, n.9, p.508-527, 2020.

MALLAT, Stéphane**. A Wavelet Tour of Signal Processing: The Sparse Way**. 3rd ed. Burlington: Academic Press, 2009.

MATURANA, Daniel; SCHERER, Sebastian. Voxnet: A 3d convolutional neural network for real-time object recognition. **2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)**. IEEE, p. 922-928,2015.

MCCULLOCH, W., PITTS, W. A Logical Calculus of the Ideas Immanent in Nervous Activity". **The Bulletin of Mathematical Biophysics**, v. 5, n. 4, p. 115 – 133, 1943

MINITAB, Inc**.** Minitab: Statistical Software. Versão 21. State College: Minitab, Inc., 2023. Disponível em: https://www.minitab.com

MONTGOMERY, Douglas C. **Design and analysis of experiments**. John Wiley & Sons, 2017.

MORTENSON, Michael E. **Mathematics for computer graphics applications**. Industrial Press Inc., 1999.

MUNSON, Bruce R.; YOUNG, Donald F.; OKIISHI, Theodore H. Fundamentals of fluid mechanics. **Oceanographic Literature Review**, v.10, n.42,p. 831,1995.

OPEN SOURCE COMPUTER VISION LIBRARY**.** OpenCV: Open Source Computer Vision Library. Versão 4.5.0. Redwood City: OpenCV.org, 2021. Disponível em: https://opencv.org/

PEDRINI, Hélio; SCHWARTZ, William Robson. **Análise de imagens digitais: princípios, algoritmos e aplicações.** Cengage Learning, 2008.

PEHERSTORFER, Benjamin; BERAN, Philip S.; WILLCOX, Karen E. Multifidelity Monte Carlo estimation for large-scale uncertainty propagation. **2018 AIAA Non-Deterministic Approaches Conference**, p 1660, 2018.

QI, Charles R. et al. Pointnet: Deep learning on point sets for 3d classification and segmentation. **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, p. 652-660, 2017.

QI, Charles Ruizhongtai et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. **Advances in Neural Information Processing Systems**, p. 5105-5114, 2017.

QIAN, Zhiguang, et al. Building surrogate models based on detailed and approximate simulations.**Journal of Mechanical Design**, p.128, 2006.

RADFORD, Alec; METZ, Luke; CHINTALA, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. **arXiv preprint arXiv:1511.06434**, 2015.

ROSENBLATT, Frank. x. **Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms**. Spartan Books, Washington DC, 1961.

RUMELHART, David E.; HINTON, Geoffrey E.; WILLIAMS, Ronald J. Learning representations by back-propagating errors. **Nature**, v.323, n.6088, p.533-536,1986.

SRIVASTAVA, Nitish, et al. Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, v.15, n.1, p.1929-1958,2014.

STANDARD, **Data Encryption**. National Institute of Standards and Technology. Federal Information Processing Standard (FIPS) Publication, v.46, n.1,2002

SUSSKIND, J., Anderson, A.; and Hinton, G. E. The Toronto face dataset. **Technical Report UTML**, U. Toronto:2010.

UMETANI, Nobuyuki; BICKEL, Bernd. Learning three-dimensional flow for interactive aerodynamic design. **ACM Transactions on Graphics (TOG)**, v. 37, n. 4, p. 1-10, 2018.

VERSTEEG, Henk K.; MALALASEKERA, Weeratunge. **Computational fluid dynamics: The finite volume method**, p.1-26,1995.

WANG, Yuqi, et al. Dual-convolutional neural network based aerodynamic prediction and multi-objective optimization of a compact turbine rotor. **Aerospace Science and Technology**, v.116, n.106869,2021.

WINTERLE, Paulo; STEINBRUCH, Alfredo. **Geometria Analítica**. Makron Books, São Paulo, 2000.

WU, Jiajun et al. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. **Advances in Neural Information Processing Systems**, n.29, 2016.

WU, Zhirong et al. 3d shapenets: A deep representation for volumetric shapes. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.** p. 1912-1920,2015.

WU, Neil, et al. pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems. **Journal of Open Source Software**, v.5, n.54 p.2564,2020.