

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

EVERTON SCHNEIDER DOS SANTOS

**APLICAÇÃO DE TÉCNICAS DE DEEP LEARNING PARA ESTIMAR O
COEFICIENTE DE UNIFORMIDADE DE MICROASPERSORES EM SISTEMAS DE
IRRIGAÇÃO POR ASPERSÃO**

Medianeira

2021

EVERTON SCHNEIDER DOS SANTOS

**APLICAÇÃO DE TÉCNICAS DE DEEP LEARNING PARA ESTIMAR O
COEFICIENTE DE UNIFORMIDADE DE MICROASPERSORES EM SISTEMAS DE
IRRIGAÇÃO POR ASPERSÃO**

**DEEP LEARNING TECHNIQUES APPLIED TO THE THE ESTIMATION OF
UNIFORMITY COEFFICIENT IN SPRINKLER IRRIGATION SYSTEMS**

Dissertação apresentada como requisito para a obtenção do título de Mestre em Tecnologias Computacionais para o Agronegócio da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Paulo Lopes de Menezes

Coorientador: Arnaldo Candido Junior

Medianeira

2021



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



TERMO DE APROVAÇÃO

Aplicação de técnicas de Deep Learning para estimar o coeficiente de uniformidade de microaspersores em sistemas de irrigação por aspersão

Por

Everton Schneider dos Santos

Esta dissertação foi apresentada no dia 27 de agosto de 2021 como requisito parcial para a obtenção do título de Mestre em Tecnologias Computacionais para o Agronegócio, da Universidade Tecnológica Federal do Paraná, Campus Medianeira. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Paulo Lopes de Menezes
UTFPR – Campus Medianeira
(Orientador)

Prof. Dr. Arnaldo Candido Junior
UTFPR – Campus Medianeira
(Co-orientador)

Prof. Dr. André Sandmann
UTFPR – Campus Medianeira

Prof Dr. Everton Coimbra de Araújo
UTFPR – Campus Medianeira

Prof^a. Dr^a. Vera Antunes de Lima
UFCG – Membro Externo

Prof^a. Dr^a. Fabiana Schütz
UTFPR – Campus Medianeira
(Coordenadora)

RESUMO

Incertezas relacionadas à disponibilidade e o preço dos recursos hídricos, causadas pelo aumento da população e mudanças climáticas, são alguns dos desafios enfrentados pela agricultura irrigada. A otimização dos processos de irrigação deverá ser feita para que ocorra uma diminuição na utilização da água pela agricultura sem que isso interfira na produção de alimentos. A simulação da uniformidade de distribuição, que é uma das principais formas de avaliação da qualidade de utilização da água em sistemas de irrigação por microaspersão, pode ajudar a diminuir os gastos com testes de campo aplicados na medição dessa variável. Este trabalho se propôs a utilizar redes neurais artificiais e técnicas de Deep Learning para realizar a predição da precipitação da água e estimar, de forma efetiva, o coeficiente de uniformidade de microaspersores. Utilizando técnicas de busca bayesiana e otimização de hiperparâmetros, um modelo de predição desenvolvido foi capaz de estimar o coeficiente de uniformidade de Christiansen, em uma distância entre aspersores de 12x12 metros, com um erro médio de apenas 0,79% no conjunto de dados de validação. Este mesmo modelo alcançou um RMSE de 64,2487 e um R^2 de 0,9287, demonstrando que a metodologia aplicada neste trabalho é capaz de simular processos de irrigação com microaspersores, podendo assim ser utilizada para diminuir os gastos relacionados à experimentação e ajudar no apoio a tomada de decisões por parte dos gerenciadores de sistemas de irrigação.

Palavras-chave: agricultura irrigada, perfil de distribuição; redes neurais artificiais; seleção de hiperparâmetros, busca bayesiana.

ABSTRACT

Uncertainties related to the availability and price of water resources caused by population growth and climate changes are some of the challenges faced by irrigated agriculture. The optimization of irrigation processes must be done so that there is a decrease in water usage by agriculture without interfering with food production. The simulation of distribution uniformity, which is one of the main ways to assess the quality of water usage in micro-sprinkler irrigation systems, can help reduce the cost of field tests used to measure this variable. This work used artificial neural networks and Deep Learning techniques to predict the water precipitation and effectively estimate the uniformity coefficient of micro-sprinklers. Using bayesian search and hyperparameter optimization, the predictive model created was able to estimate Christiansen's uniformity coefficient with an average error of 0,79% on the validation set. The same model achieved a RMSE of 64,2487 and a R^2 of 0,9287, showing that the methodology used in this work is capable of simulate micro-sprinklers irrigation processes, used to reduce costs associated with experimentation and support the decision-making process made by irrigation system managers.

Key-words: irrigated agriculture, distribution profile, artificial neural networks, hyperparameter selection, bayesian search.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 - Microaspersor rotativo com sistema de dispersão a jato | 16 |
| Figura 2 - Perfis de Distribuição de Christiansen | 17 |
| Figura 3 - Arquitetura de um neurônio..... | 23 |
| Figura 4 - Arquitetura de uma rede <i>feed-forward</i> | 24 |
| Figura 5 - Gradiente Descendente | 26 |
| Figura 6 - Gráfico da função sigmoide | 27 |
| Figura 7 - Gráfico da função tanh..... | 28 |
| Figura 8 - Fluxograma do processo de desenvolvimento | 36 |
| Figura 9 - Mapa de localização da fazenda Ouro Verde | 37 |
| Figura 10 - Disposição dos pluviômetros em torno do microaspersor..... | 38 |
| Figura 11 - Fluxograma do processo de pipeline | 45 |
| Figura 12 - Conjunto de dados pré-processado | 47 |
| Figura 13 – MAE e MSE para os conjuntos de treinamento e validação do melhor modelo | 51 |
| Figura 14 – MAE e MSE dos conjuntos de treinamento e validação de um modelo com <i>overfitting</i> | 51 |
| Figura 15 - Página de realização de previsões do ambiente <i>web</i> | 53 |
| Figura 16 - Comparação da superfície de precipitação dos valores reais e dos valores preditos | 53 |
| Figura 17 - Comparação do perfil de distribuição dos valores reais e dos valores preditos | 54 |

LISTA DE TABELAS

| | |
|--|---------|
| Tabela 1 - Critérios de desempenho dos coeficientes de uniformidade de irrigação | 19 |
| Tabela 2 - Conjuntos de dados utilizados como entrada dos modelos de predição | ..48 |
| Tabela 3 - Resultados de predição alcançados pelo modelo de uma camada |49 |
| Tabela 4 - Melhores hiperparâmetros encontrados para o modelo de uma camada | 49 |
| Tabela 5 - Resultados de predição alcançados pelos modelos de várias camadas | ..50 |
| Tabela 6 - Comparação entre os valores reais e os valores preditos de CUC |52 |

SUMÁRIO

| | |
|--|-----------|
| 1 INTRODUÇÃO | 9 |
| 1.1 JUSTIFICATIVA | 10 |
| 1.2 OBJETIVOS | 12 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 13 |
| 2.1 SISTEMAS DE IRRIGAÇÃO | 14 |
| 2.1.1 IRRIGAÇÃO POR MICROASPERSÃO | 15 |
| 2.1.2 PERFIL DE DISTRIBUIÇÃO | 16 |
| 2.1.3 UNIFORMIDADE DE DISTRIBUIÇÃO | 17 |
| 2.2 APRENDIZADO DE MÁQUINA | 19 |
| 2.3 REDES NEURAIS ARTIFICIAIS | 22 |
| 2.3.1 ARQUITETURA DE UM PERCEPTRON | 23 |
| 2.3.2 ESTRUTURAS DE REDES NEURAIS | 24 |
| 2.3.3 O ALGORITMO <i>BACKPROPAGATION</i> | 25 |
| 2.3.4 GRADIENTE DESCENDENTE | 26 |
| 2.3.5 FUNÇÕES DE ATIVAÇÃO | 27 |
| 2.3.6 OTIMIZAÇÃO DE HIPERPARÂMETROS | 29 |
| 2.4 AVALIAÇÃO ESTATÍSTICA | 30 |
| 2.4.1 NORMALIZAÇÃO | 31 |
| 2.4.2 GENERALIZAÇÃO DE MODELOS | 32 |
| 2.4.3 TÉCNICAS DE DEEP LEARNING | 33 |
| 3 MATERIAIS E MÉTODOS | 34 |
| 3.1 AMBIENTE DE DESENVOLVIMENTO | 35 |
| 3.2 PROCESSO DE DESENVOLVIMENTO | 35 |
| 3.3 CONJUNTO DE DADOS | 37 |
| 3.4 PRÉ-PROCESSAMENTO E ANÁLISE DOS DADOS | 38 |
| 3.5 SELEÇÃO DE VARIÁVEIS E DIVISÃO DOS DADOS | 39 |
| 3.6 DESENVOLVIMENTO DOS MODELOS | 40 |
| 3.7 SELEÇÃO DE HIPERPARÂMETROS | 42 |
| 3.8 PIPELINE | 44 |
| 3.9 ANÁLISE ESTATÍSTICA DOS RESULTADOS | 45 |
| 3.10 AMBIENTE <i>WEB</i> | 46 |
| 4 RESULTADOS | 47 |
| 4.1 VISÃO GERAL DOS RESULTADOS – MODELO UMA CAMADA | 48 |
| 4.2 VISÃO GERAL DOS RESULTADOS – MODELO VÁRIAS CAMADAS | 50 |
| 4.3 ANÁLISE DO RESULTADOS – MELHOR MODELO | 50 |
| 4.4 DISCUSSÃO DOS RESULTADOS | 54 |
| 5 CONCLUSÃO | 56 |
| 5.1 TRABALHOS FUTUROS | 57 |
| REFERÊNCIAS | 58 |

1 INTRODUÇÃO

A segurança alimentar existe, segundo Food and Agriculture Organization of the United Nations (2009), quando todas as pessoas possuem acesso físico, social e econômico a uma alimentação segura, nutricional, capaz de atender suas necessidades e preferências alimentares para uma vida ativa e saudável.

Os processos de irrigação contribuem diretamente para a segurança alimentar por meio do aumento da produção e da variedade agrícola nas mais diversas formas. Além disso, áreas importantes como a indústria de biocombustíveis são impactadas de forma positiva pelas melhorias que o desenvolvimento correto de um sistema de irrigação pode trazer para a agricultura. Seja pelo aumento da produtividade ou pela expansão da área de cultivo quando a insuficiência de chuva limita a área plantada, os mais diversos aspectos da produção agrícola são afetados favoravelmente pela irrigação (PANDEY, 2019).

Estes motivos tornam a agricultura um ramo importante para a sociedade como um todo. A capacidade de realização do processo de irrigação de forma inteligente e a criação de perfis de produção de alimentos que minimizem o desperdício de água são questões centrais no estudo dos sistemas de irrigação.

Isso acontece devido a mudanças no perfil da população mundial, aumento do consumo de insumos e *commodities* em países em desenvolvimento, além de mudanças climáticas que afetam a quantidade e a qualidade da produção agrícola. É nesse contexto que a busca de novas soluções para resolver esses problemas ganha destaque.

A difusão de sistemas de irrigação modernos, como os sistemas de microaspersão, trouxeram avanços importantes para o setor devido a sua facilidade de instalação e capacidade de diminuição do desperdício de água. Porém, devido à alta influência que fatores meteorológicos possuem nesse tipo de sistema, questionamentos importantes devem ser levados em consideração, como a escolha de características do microaspersor que garantam uma maior eficiência na uniformidade de distribuição da água.

A evolução da tecnologia da informação ajudou a solucionar os mais diversos problemas em áreas como medicina, arquitetura, negócios e educação. Atualmente, as soluções baseadas em aprendizado de máquina vêm ganhando

destaque pela sua capacidade de gerar conhecimento utilizando a grande quantidade de dados gerada nas mais diferentes áreas de atuação.

Essa atual conjuntura acaba gerando questionamentos relacionados à capacidade que avanços computacionais possuem para melhorar os processos relacionados aos sistemas de irrigação, a viabilidade que as técnicas de aprendizado de máquina possuem como soluções aos problemas da agricultura irrigada e a possibilidade de criação de modelos de inteligência artificial para utilização de recursos hídricos em sistemas de irrigação.

Este trabalho de dissertação nasceu da necessidade de encontrar respostas para questões como: técnicas de aprendizado de máquina são capazes de serem aplicadas na melhoria dos processos de irrigação? Redes neurais artificiais podem ser utilizadas para realizar a estimativa satisfatória da quantidade de água utilizada na agricultura irrigada? Existem diferenças nos resultados alcançados com a utilização de técnicas tradicionais de aprendizado de máquina e novas tecnologias desenvolvidas na área de redes neurais artificiais? Por meio da aplicação de uma pesquisa experimental, espera-se encontrar respostas satisfatórias para esses questionamentos.

O presente trabalho está dividido da seguinte maneira: inicialmente são apresentadas as justificativas, descrição do estado da arte da pesquisa desenvolvida e os objetivos esperados. Após isso uma revisão da literatura das áreas relevantes ao trabalho executado é realizada: agricultura irrigada, aprendizado de máquina, redes neurais artificiais e novas técnicas dessa área, finalizando com a fundamentação dos métodos estatísticos aplicados aos resultados da pesquisa. Após isso é realizada uma descrição dos materiais e métodos aplicados na execução do trabalho e a análise e discussão dos resultados alcançados.

1.1 JUSTIFICATIVA

Em Food and Agriculture Organization (2017) é definida como questão central para o futuro da agricultura a capacidade de atender as demandas alimentares de uma população em crescimento de maneira sustentável já que o crescimento econômico e a expansão da produção alimentar têm um alto custo ambiental.

Nesse cenário, é importante encontrar novas maneiras sustentáveis de aumentar a produção de alimentos e que, ao mesmo tempo, minimizem os impactos ambientais causados por esse aumento. Avanços tecnológicos e o aumento expressivo na geração de dados, tanto no campo quanto na cidade, tornam as ferramentas computacionais, especialmente as baseadas em aprendizado de máquina, candidatos ideais para tentar solucionar esses problemas.

Diversos estudos voltados à melhoria dos processos de irrigação na agricultura foram desenvolvidos recentemente, com destaque para os trabalhos que utilizaram soluções de aprendizado de máquina para alcançar um consumo inteligente dos recursos hídricos.

Sayari, Mahdavi-Meymand e Zounemat-Kermani (2021) alcançaram bons resultados ao utilizar técnicas de aprendizado de máquina para realizar a predição de infiltração da água em sistemas de irrigação por sulcos. Já Wilfrand (2018) aplicou uma abordagem teórica-conceitual na aplicação de redes neurais artificiais para estimar a precipitação de água de aspersores. Modelos de inteligência artificial foram aplicados com sucesso por El Bilabi, Taleb e Brouzuiyne (2021) na estimativa da qualidade da água de lençóis freáticos utilizados na irrigação.

Outro fator importante para o gerenciamento de sistemas de irrigação é a uniformidade de distribuição da água. Mas, apesar dessa importância, o estudo da utilização de técnicas de aprendizado de máquina na modelagem dessa variável ainda é limitado, especialmente se comparado a outros fatores como evapotranspiração e umidade do solo.

Uma simples busca em plataformas especializadas no compartilhamento de artigos científicos, como o ScienceDirect, demonstra bem essa diferença. Uma busca pelos termos *irrigation* e *evapotranspiration* retorna mais 25 mil resultados enquanto a busca por *soil moisture* e *irrigation* resulta em mais de 30 mil artigos. Já a pesquisa por trabalhos contendo as palavras *irrigation* e *distribution uniformity* retorna menos de 500 artigos.

O presente trabalho pretende ajudar a preencher essa lacuna, demonstrando a capacidade que as técnicas de aprendizado de máquina, especialmente as novas técnicas baseadas em redes neurais artificiais, possuem na estimativa de precipitação da água em sistemas de irrigação por microaspersão. Isso poderá ajudar a agricultura irrigada a alcançar um consumo de água realmente

sustentável, diminuindo os impactos ambientais causados pelo aumento da produção de alimentos.

1.2 OBJETIVOS

O objetivo geral deste trabalho é desenvolver um modelo baseado em redes neurais que estime, de forma efetiva, o coeficiente de uniformidade de distribuição da água de microaspersores utilizados na agricultura irrigada, que minimize a utilização dos recursos hídricos sem interferir na produção agrícola.

Entre os objetivos específicos desta pesquisa podem ser citados:

- Desenvolver modelos de redes neurais que sejam capazes de estimar o coeficiente de uniformidade de microaspersores;
- Encontrar o melhor conjunto de hiperparâmetros para cada um dos modelos desenvolvidos;
- Realizar a avaliação, por meio de métodos estatísticos, dos modelos desenvolvidos e seus respectivos conjuntos de hiperparâmetros.
- Desenvolver uma aplicação *web* para apresentar os resultados do trabalho e ser utilizada como base de apoio a tomada de decisões.

2 FUNDAMENTAÇÃO TEÓRICA

A agricultura é uma área de importância para a humanidade, já que este ramo de atividade utiliza água (irrigação) para a produzir a matéria-prima necessária para a produção de alimentos. Mudanças climáticas e o rápido aumento da população colocam pressão na agricultura, o que acaba gerando um efeito bola de neve nos recursos hídricos do planeta, que são cruciais para um desenvolvimento sustentável (ADENUGBA et al., 2019).

Apesar da agricultura ser o maior consumidor de água captada no mundo, o desperdício desse recurso tão importante é muito grande. Atualmente, metade dos 3.600km³ de água potável são perdidos devido a fatores como a evaporação e transpiração da colheita, enquanto a outra metade é adicionada aos lençóis freáticos (NAWANDAR; SATPUTE, 2019).

Esse aumento constante da utilização de água por parte da agricultura, mudanças no clima e uma demanda maior de consumo por parte da população, levaram a uma grande redução na disponibilidade de água potável. Além disso, incertezas associadas a demanda e o aumento do custo da água são desafios enfrentados diariamente no gerenciamento de sistemas de irrigação (PEREA et al., 2018).

Setores produtivos, incluindo a agricultura, são responsáveis pela utilização da maior parte da água potável no mundo e existe uma demanda cada vez maior para a produção de alimentos e para o aumento da produção de biocombustíveis, sendo que a maior parte dessa produção vem do setor agrícola (GARCÍA-SALDANA et al., 2019).

Como a irrigação é uma prática importante em diversos tipos de colheita, formas úteis de gerenciamento e aplicação da água são preocupações chave para a agricultura. A eficiência e uniformidade da irrigação podem ser mantidas por sistemas de informações que utilizam dados meteorológicos, do solo, água e da cultura (MOHAPATRA; LENKA; KESWANI, 2019).

A melhoria na eficiência e produtividade do uso da água é considerada indispensável para fornecer a água necessária para o consumo humano e para as atividades industriais. A limitação da disponibilidade dos recursos hídricos torna a

maximização da renda líquida por unidade de água um requisito para uma agricultura sustentável (VOLSCHENK, 2020).

A sustentabilidade da agricultura irrigada está fortemente ligada as melhorias na eficiência do uso da água. A estimativa da demanda de água, por exemplo, pode se tornar uma das principais ferramentas para desenvolver, de forma precisa, novos sistemas de irrigação e melhorar o gerenciamento de antigas redes de irrigação (GONZÁLEZ et al., 2019).

Um novo tipo de tecnologia que vem ganhando destaque pela sua capacidade de prover soluções para problemas nas mais diversas áreas, inclusive na agricultura, são as soluções baseadas em aprendizado de máquina, especialmente as técnicas que utilizam redes neurais artificiais.

Algoritmos de aprendizado de máquina foram utilizados por Meroni et al. (2021) na realização da predição do rendimento mensal de colheitas entre o início e o fim da temporada de cultivo. Já Danner *et al.* (2021) aplicaram redes neurais artificiais e outras técnicas de aprendizado de máquina na quantificação de características biofísicas e bioquímicas de culturas a partir de imagens espectroscópicas.

Estes são apenas alguns dos exemplos que demonstram a capacidade que esse tipo de técnica possui para solucionar problemas em diferentes áreas da agricultura e que elas podem, inclusive, ajudar a responder questões relacionadas a agricultura irrigada e contribuir para o alcance da segurança alimentar.

2.1 SISTEMAS DE IRRIGAÇÃO

Dos requerimentos básicos para a produção agrícola, a água é o mais importante para o crescimento da planta, sendo responsável pela umidade necessária para a germinação de plantas, resfriamento do solo e aplicação de fertilizantes. Porém, nem sempre a água vinda das chuvas ou do subsolo é suficiente para que uma planta se desenvolva. Por isso, é necessária a aplicação de água para compensar essa deficiência. Essa aplicação artificial de água para suplementar a umidade natural do solo para o propósito de produção agrícola é conhecida como irrigação (ASAWA, 2008).

Os sistemas de irrigação foram amplamente desenvolvidos ao redor do mundo devido a sua capacidade de aumentar a produtividade da colheita de forma significativa. Porém, devido a utilização de grandes volumes de água e a possibilidade de contaminação de corpos d'água, uma atenção especial deve ser dada ao gerenciamento de água em terras irrigadas. A água deve ser aplicada na quantidade correta e na hora correta, com o objetivo de otimizar a produção, diminuir o desperdício desse recurso valioso e aplicá-lo de forma ambientalmente sustentável (KARANDISH; SIMUNEK, 2016).

Testezlaf (2017) define a irrigação como uma série de técnicas utilizadas na aplicação artificial da água em plantas, com o objetivo de satisfazer as necessidades e alcançar uma produção ideal para o usuário e que é necessário pensar a irrigação como uma ciência, buscando formas de otimizar a utilização da água e diminuir o desperdício.

Para que isso aconteça, se faz necessária a escolha de um método de aplicação que resulte em uma utilização otimizada da água disponível. Essa escolha deve ser baseada em considerações como fonte e custo da água, sua quantidade e qualidade, os efeitos no ambiente, energia, equipamento, operação e manutenção (USDA, 1997).

Entre os principais tipos de sistemas de irrigação podem ser citados a irrigação por superfície, por aspersão, a irrigação localizada e a de subsuperfície.

2.1.1 IRRIGAÇÃO POR MICROASPERSÃO

O sistema de irrigação por microaspersão é um tipo de irrigação localizada que utiliza pequenos emissores que lançam jatos de água no ar e que viajam pequenas distâncias até chegar no solo. Em comparação com outros tipos de sistemas, a microaspersão possui uma capacidade muito maior de irrigação devido a sua vazão superior (TESTEZLAF, 2017).

Os microaspersores são emissores de água de baixa capacidade e com um tamanho menor quando comparados aos aspersores tradicionais. Os sistemas que utilizam essa ferramenta são sazonais, de baixa pressão e de fácil instalação e remoção. Entre as vantagens desse sistema podem ser citadas sua flexibilidade e adaptabilidade, facilidade de adoção e gerenciamento (PHOCAIDES, 2007).

A irrigação por microaspersão utiliza dois tipos principais de microaspersores: o tipo rotativo e o tipo estacionário. O microaspersor do tipo rotativo, que pode ser visto na Figura 1, a distribuição da água é feita por um sistema rotativo de dispersão. Já no tipo estacionário, uma placa defletora irá determinar a forma como a água é dispersa na área a ser irrigada (TESTEZLAF, 2017).

Figura 1 - Microaspersor rotativo com sistema de dispersão a jato



Fonte: Antunes (2006)




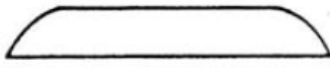
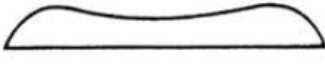
Um sistema projetado de forma correta aborda a aplicação uniforme da irrigação em tempo hábil, minimizando perdas e danos ao solo, plantas e outros recursos. Sistemas de irrigação devem ser capazes de fornecer e aplicar a quantidade de água necessária para atender a demanda de água da cultura. Além disso, as taxas de aplicação não devem exceder a taxa de infiltração máxima permitida para cada tipo de solo (SORRENTINO; PÉREZ, 2005).

2.1.2 PERFIL DE DISTRIBUIÇÃO

Keller e Bliesner (1990) afirmam que a escolha de um aspersor tem como objetivo encontrar uma combinação de espaçamento entre aspersores, pressão e tamanho do bocal que irá alcançar uma taxa de aplicação de água ótima com o maior grau de uniformidade de distribuição.

Chstiansen (1942) define cinco perfis estilizados de precipitação gerados por aspersores operando em ventos fracos. Estes perfis, apresentados na Figura 2, determinam o formato da curva radial de cada perfil, além do espaçamento entre aspersores em uma distribuição de formato quadrado.

Figura 2 - Perfis de Distribuição de Christiansen

| Perfil | Forma | Espaçamento |
|--------|--|-------------|
| A |  | 50 |
| B |  | 55 |
| C |  | 60 |
| D |  | 40 |
| E |  | 40 |

Fonte: Adaptado de Faria (2011)

Segundo Keller e Bliesner (1990), os perfis A e B são característicos de aspersores com dois ou mais bocais, enquanto os perfis C e D são típicos de um aspersor com um único bocal na pressão recomendada. Já o perfil E geralmente é produzido por um aspersor de arma ou de aspersores operando em pressões abaixo do recomendado para o tamanho do bocal.

Modelos de simulação de irrigação foram desenvolvidos para evitar testes de campo trabalhosos e para projetar e melhorar sistemas de irrigação. A capacidade de modelar, de forma precisa, a uniformidade de distribuição em sistemas de irrigação possui um papel importante na utilização otimizada da água na irrigação, com repercussões diretas na aplicação eficiente da água e na produção (MONTERO; TARJUELO; CARRIÓN, 2001).

2.1.3 UNIFORMIDADE DE DISTRIBUIÇÃO

A modernização dos processos executados em áreas irrigadas levou a instalação de redes pressurizadas. Isso acarretou em uma mudança de paradigma, onde os produtores passaram a usar irrigação por gotejamento ou aspersão ao invés

da tradicional irrigação por superfície. Esses novos métodos possuem uma maior eficiência na aplicação da água, melhor controle da profundidade da água aplicada e permite a automação do sistema (BELLIDO; MARTÍNEZ; URRESTARAZU, 2019).

Apesar disso, a irrigação por aspersão é altamente influenciada por variáveis meteorológicas, como a velocidade e direção do vento, que causam perdas parciais de água por meio da evaporação e do desvio para fora da área de irrigação. Isso causa uma distorção e estreitamento no padrão de distribuição da água em aspersores isolados (SHEIKHESMAEILI; MONTERO; LASERNA, 2016).

A uniformidade de irrigação, ou uniformidade de distribuição da água, é uma das características de desempenho mais importantes em sistemas de irrigação por aspersão. Esse fator de planejamento afeta aspectos importantes como o uso eficiente da água, lixiviação de fertilizantes e o rendimento de colheitas (SEGINER; NIR; BERNUTH, 1991).

Métodos como o coeficiente de distribuição de uniformidade e o coeficiente de uniformidade de Christiansen (CUC) são utilizados para expressar a uniformidade de aplicação da água, considerado o principal indicador de qualidade da irrigação (BELLIDO; MARTÍNEZ; URRESTARAZU, 2019).

Na fórmula de medição do coeficiente de Christiansen, apresentada na Equação 1, o x_i representa a profundidade da água em cada coletor utilizado para medição, \bar{x} representada a média dos valores de profundidade da água em todos os coletores e o n indica a quantidade de coletores utilizados para realizar a medição (MAROUFPOOR; SHIRI; MAROUFPOOR, 2019).

$$\left(1 - \sum_{i=1}^n \frac{|x_i - \bar{x}|}{n\bar{x}} \right) \times 100 \quad (1)$$

Outro método de avaliação da uniformidade da irrigação é através do cálculo do coeficiente de uniformidade de distribuição (CUD), apresentado na Equação 2. Nessa fórmula o $\bar{x}_{25\%}$ representa a média de 25% do total de coletores com as menores vazões enquanto o \bar{x} representa a média da vazão dos coletores da subárea.

$$\frac{\bar{x}25\%}{\bar{x}} \times 100 \quad (2)$$

A Equação 3 apresenta o coeficiente de uniformidade de emissão (CUE0, outra metodologia empregada na avaliação da uniformidade da irrigação. Nessa fórmula o σ representa o desvio padrão da vazão dos coletores e o \bar{x} representa a medida da vazão dos coletores da subárea.

$$\left(1 - \frac{\sigma}{\bar{x}}\right) \times 100 \quad (3)$$

Segundo a metodologia definida em Alves et al. (2015), a Tabela 1 apresenta a classificação de desempenho para cada um dos coeficientes apresentados.

Tabela 1 - Critérios de desempenho dos coeficientes de uniformidade de irrigação

| Classificação | CUC (%) | CUD (%) | CUE (%) |
|----------------------|----------------|----------------|----------------|
| Excelente | > 90 | > 84 | 90 – 100 |
| Bom | 80 – 90 | 68 - 84 | 80 – 90 |
| Razoável | 70 – 80 | 52 – 68 | 70 – 80 |
| Ruim | 60 – 70 | 36 – 52 | 60 – 70 |
| Inaceitável | < 60 | < 36 | < 60 |

Fonte: Mantovani (2001) apud Alves et al. (2015)

Segundo Keller e Bliesner (1990), a maioria dos sistemas de irrigação por aspersão exigem um valor mínimo do coeficiente de Christiansen maior que 80%. Bralts et al. (1994) afirmam que um aumento de 5 a 12% no valor desse parâmetro pode aumentar o rendimento da colheita de grão de trigo em até 17%.

2.2 APRENDIZADO DE MÁQUINA

A Inteligência Artificial (IA) é um ramo da ciência da computação que estuda a utilização de computadores na simulação de processos de inteligência humana como aprendizagem, raciocínio, resolução de problemas e autocorreção. A criação de máquinas que funcionam, reagem e copiam funções cognitivas de humanos é o objetivo principal da área de IA (BELYADI; HAGHIGHAT, 2021)

O aprendizado de máquina pode ser definido, segundo Schalkoff (1990), como um campo de estudo que busca explicar e emular um comportamento inteligente em termos de processos computacionais. Já Luger e Stubblefield (1993) definem aprendizado de máquina como o ramo da ciência da computação que se preocupa com a automação de comportamentos inteligentes.

De forma ampla, o aprendizado de máquina pode ser descrito como um conjunto de métodos computacionais que utilizam experiências passadas para melhorar o desempenho ou realizar previsões. O aprendizado de máquina consiste no desenvolvimento de algoritmos que realizem essas previsões de forma eficiente e precisa (MOHRI; ROSTAMIZADEH; TALWALKAR, 2012).

O aprendizado de máquina é uma ferramenta que se popularizou na execução de tarefas que necessitam extrair informações de grandes conjuntos de dados. Uma característica compartilhada por essas aplicações é o fato de que um programador não é capaz de fornecer uma especificação detalhada de como tais tarefas devem ser executadas. Isso acontece devido à complexidade dos padrões que precisam ser detectados (SHALEV-SHWARTZ; BEN-DAVID, 2014).

Apesar dessa complexidade, avanços tecnológicos na área permitiram o desenvolvimento de diversas ferramentas de aprendizado de máquina que foram aplicadas com sucesso em áreas como de mineração de dados, detecção de fraudes, reconhecimento facial, processamento de linguagem natural, entre outros.

Entre os avanços tecnológicos que permitiram a implementação bem-sucedida dessas ferramentas pode ser dado destaque a recursos computacionais cada vez mais poderosos e baratos, que possibilitaram o processamento de conjuntos de dados cada vez maiores, gerados pela profunda integração de dispositivos computacionais nos mais diversos aspectos do dia-a-dia da população mundial.

Uma das principais vantagens das técnicas de aprendizagem de máquina é a capacidade de solucionar, de forma autônoma, problemas não lineares por meio da utilização de conjuntos de dados vindos de múltiplas fontes. Essas técnicas garantem a melhoria na tomada de decisão e permitem a tomada de ações informadas em cenários reais sem a intervenção humana. O aprendizado de máquina fornece uma estrutura poderosa e flexível para uma tomada de decisão baseada em dados e incorpora no sistema um conhecimento especialista (CHLINGARYAN; SUKKARIEH; WHELAN, 2018).

Russel e Norvig (1995) afirmam que um agente está aprendendo se o seu desempenho na realização de tarefas futuras está melhorando após a realização de observações sobre o mundo. Ainda segundo Russel e Norvig (1995), qualquer componente desse agente pode ser melhorado pelo aprendizado com dados. Essas melhorias e as técnicas utilizadas para realizá-las dependem do componente a ser aprimorado, o conhecimento passado que o agente já possui, a representação utilizada e qual o tipo de resposta pode ser aprendida. Baseado no tipo de resposta, existem três principais formas de aprendizagem: o aprendizado supervisionado, o aprendizado não supervisionado e o aprendizado por reforço.

O aprendizado supervisionado é definido por Vanderplas (2017) como o processo de modelagem do relacionamento entre características quantificadas do conjunto de dados e um rótulo associado. Assim que o modelo é determinado, este pode ser utilizado na aplicação de rótulos em novos dados que ainda não foram observados.

Já no aprendizado não supervisionado, não existem rótulos e a qualidade da representação que um modelo necessita aprender é mensurada por meio de uma medida de independência da tarefa. Assim que o modelo é ajustado em relação aos dados, este é capaz de formar representações internas das características de entrada e, deste modo, criar novos rótulos de forma automática (HAYKIN, 1999).

O aprendizado por reforço é uma técnica de aprendizagem que direciona uma atividade à maximizar a recompensa de uma ação imediata e daquelas que a seguem. Neste tipo de algoritmo, um modelo treina a si mesmo de forma contínua utilizando uma abordagem computacional onde ele é capaz de aprender com a execução e o resultado de uma ação (BELYADI; HAGHIGHAT, 2021).

Entre as principais tarefas do aprendizado supervisionado estão a classificação e a regressão. Os modelos de classificação observam um conjunto de características para realizar a predição de uma categoria, dentro de um conjunto discreto de opções, na qual um determinado exemplo pertence. De forma geral, um modelo de classificação busca criar um classificador, que possuirá como saída a atribuição de classe prevista para a entrada (ZHANG *et al.*, 2021).

A regressão é um tipo de tarefa onde um programa computacional tem como objetivo realizar a predição de um valor numérico em relação a uma determinada entrada. Esse tipo de tarefa é similar à classificação, porém o formato da saída gerada é diferente. Enquanto os problemas de classificação geram uma resposta

dentro de um conjunto discreto, a regressão tem como resultado uma resposta dentro de um conjunto contínuo (GOODFELLOW; BENGUI; COURVILLE, 2016).

Apesar de sua capacidade em resolver diversos problemas, o campo do aprendizado de máquina ainda levanta diversas questões, tais como a existência de funções de aprendizado genéricas que foram geradas por exemplos específicos, como realizar a escolha do melhor tipo de algoritmo para determinados tipos de problemas e representações ou qual a quantidade suficiente de dados que devem ser utilizados (MITCHELL, 1997).

Uma técnica de aprendizado de máquina que se popularizou devido a sua capacidade de ser uma resposta efetiva para esses tipos de questionamentos foi a rede neural artificial.

2.3 REDES NEURAS ARTIFICIAIS

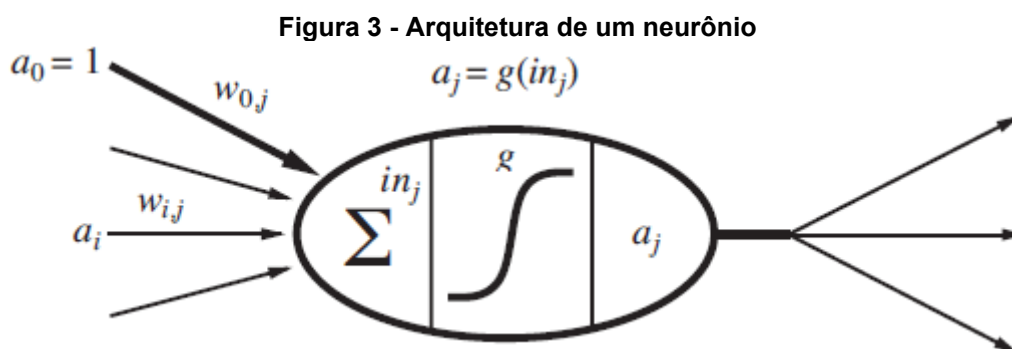
Uma rede neural artificial é um modelo computacional inspirado pela estrutura de redes neurais cerebrais. Modelos simplificados do cérebro consistem de um grande número de dispositivos computacionais básicos (neurônios), conectados através de uma complexa rede de comunicação, pela qual o cérebro é capaz de realizar computações altamente complexas (SHALEV-SHWARTZ; BEN-DAVID, 2014).

Uma rede neural artificial pode ser definida como um processador paralelo e distribuído, feito de diversas unidades de processamento simples, com uma capacidade de armazenar conhecimento e torná-lo disponível para utilização. Nesse tipo de modelo o conhecimento é adquirido por meio de um processo de aprendizagem e pela força de conexão entre as unidades (HAYKIN, 1999)

Essa força de conexão, também chamada de peso, é a principal forma de armazenamento de longo termo em redes neurais e o aprendizado normalmente acontece por meio da atualização desses pesos. A modificação desses valores tem como objetivo alinhar o comportamento de entrada e saída da rede com o do ambiente que está providenciando as entradas (RUSSELL; NORVIG, 1995).

2.3.1 ARQUITETURA DE UM PERCEPTRON

Uma rede neural artificial é composta por um número de unidades interconectadas. Uma representação típica de um desses nós é apresentada na Figura 3. Uma ligação entre as unidades i e j é utilizada para propagar a ativação a_i entre elas. Cada ligação também possui um peso numérico $w_{i,j}$ associado e que irá determinar a força e o sinal da conexão. Cada unidade também possui uma entrada simulada $a_0 = 1$ com um peso associado $w_{0,j}$. Cada unidade j irá, inicialmente computar a soma ponderada de todas as suas entradas (RUSSELL; NORVIG, 1995).



Fonte: Russel e Norvig (1995)

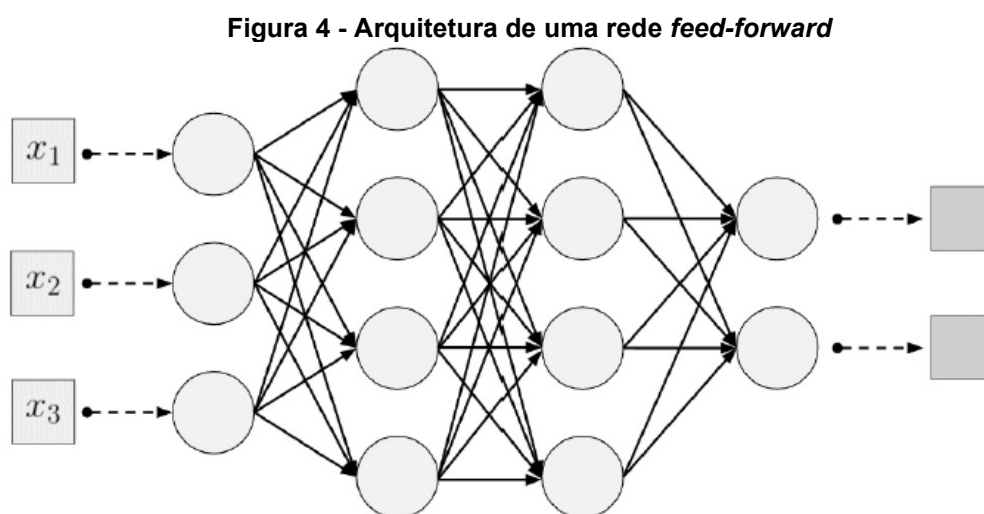
Após isso, a unidade irá aplicar uma função de ativação g na soma das entradas para então derivar a saída. A função g é normalmente um limite rígido, e nesse caso a unidade é chamada de *perceptron*, ou uma função logística. Estes dois tipos de função de ativação garantem a capacidade da rede como um todo em representar uma função não linear (RUSSELL; NORVIG, 1995).

Um único *perceptron* é a forma mais simples de uma rede neural utilizada na classificação de padrões linearmente separáveis, aqueles padrões que existem em lados opostos de um hiperplano. Esse tipo de arquitetura é limitado a desempenhar a classificação de apenas duas classes. Ao expandir a camada de saída de um *perceptron* para incluir mais de um neurônio, é possível realizar a classificação de mais de duas classes (HAYKIN, 1999).

2.3.2 ESTRUTURAS DE REDES NEURAIS

A ideia por trás das redes neurais artificiais é que diversos neurônios podem ser conectados através de elos de comunicação com o objetivo de executar computações complexas. É comum descrever a estrutura de uma rede neural como um grafo, onde os nós são os neurônios e cada aresta direcionada no grafo irá conectar a saída de um neurônio com a entrada de outro neurônio (SHALEV-SHWARTZ; BEN-DAVID, 2014).

Um dos mais populares tipos de redes neurais com múltiplas camadas é a *feed-forward neural network*. A arquitetura dessa rede, descrita na Figura 4, consiste de uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Todas as camadas são completamente conectadas e possuem um ou mais neurônios artificiais, que por sua vez podem possuir diferentes funções de ativação (PATTERSON; GIBSON, 2017).



Fonte: Patterson e Gibson (2017)

Nessa estrutura, a primeira camada recebe os dados de entrada x_1 , x_2 , x_3 a serem utilizados pela rede. Essa camada é seguida por uma ou mais camadas ocultas e elas podem ser totais ou parcialmente conectadas na próxima camada. Os pesos ponderados nas conexões são utilizados pelas redes neurais para codificar a informação aprendida. Por fim, a camada de saída gera uma resposta para o modelo. Essa resposta pode ser um valor contínuo (regressão) ou um conjunto de

probabilidades (classificação). Isso é controlado pelo tipo de função de ativação utilizada pelos neurônios de saída (PATTERSON; GIBSON, 2017).

Na estrutura de rede *feed-forward*, as conexões entre neurônios são unidirecionais e não existem ciclos, formando um grafo acíclico dirigido. Essa estrutura contrasta com as redes do tipo recorrente, onde as conexões podem formar topologias arbitrárias (RUSSELL; NORVIG, 1995).

2.3.3 O ALGORITMO *BACKPROPAGATION*

Um dos métodos de aprendizado mais utilizados em redes neurais do tipo *feed-forward* é o algoritmo *backpropagation*. Nessa implementação acontece o avanço do padrão de entrada, a retro propagação do erro associado e o ajuste dos pesos. Esse algoritmo é utilizado para dividir a contribuição de cada um dos pesos e tem como objetivo a minimização do erro entre cada saída alvo e a saída que realmente foi computada pela rede (RUSSELL; NORVIG, 1995).

O algoritmo *backpropagation* realiza o cálculo do gradiente dos parâmetros de uma rede neural. Esse método passa pela rede em ordem inversa, da camada de saída à camada de entrada, de acordo com a regra da cadeia do cálculo. O algoritmo então armazena todas as variáveis intermediárias necessárias ao calcular o gradiente em relação a um determinado conjunto de parâmetros (ZHANG *et al.*, 2021).

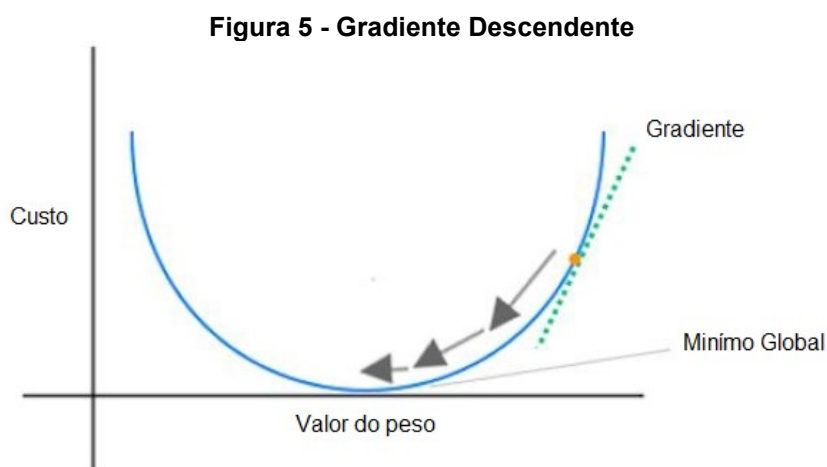
É por meio desse método que a responsabilidade pelo erro é distribuída para trás através da rede. Cada um dos nós enviando uma entrada para a unidade atual é, de alguma forma, responsável por uma parte do erro em cada um dos neurônios que ele possui uma conexão direta (PATTERSON; GIBSON, 2017).

De forma resumida, para cada instância de treinamento, o algoritmo *backpropagation* inicialmente realiza uma predição, mede o erro e só então o método passa por cada uma das camadas de forma reversa realizando a medição da contribuição para o erro em cada uma das conexões. O algoritmo então executa a correção do valor dos pesos em cada uma das conexões com o objetivo de diminuir o erro. Esse passo final é conhecido também como o gradiente descendente (GÉRON, 2019).

2.3.4 GRADIENTE DESCENDENTE

O algoritmo *backpropagation* é uma técnica utilizada para realizar a implementação do gradiente descendente no conjunto de pesos de uma rede neural de múltiplas camadas. A ideia básica é calcular de forma eficiente as derivadas parciais de uma função de aproximação realizada pela rede em relação a todos os elementos do conjunto de pesos para cada valor de entrada que irá passar pela rede (HAYKIN, 1999)

O gradiente descendente, apresentado na Figura 5, é um método popular para encontrar o valor mínimo de funções complexas e, dentro de uma rede neural, ele é utilizado no processo de atualização dos valores dos pesos. Esse procedimento encontra o caminho mais íngreme para alcançar o mínimo de uma função de custo. A inclinação é definida como o declive, ou primeira derivada, da função e ela é utilizada para avançar em direção ao valor mínimo (BELYADI; HAGHIGHAT, 2021).



Fonte: Adaptado de Patterson e Gibson (2017)

Considerando os valores dos erros como uma função dos pesos da rede, a técnica do gradiente descendente é utilizada para minimizar o erro sujeito aos pesos por meio de um avanço do tamanho do gradiente de erro. Dependendo se o erro for positivo ou negativo, o gradiente poderá se mover para à direita ou para à esquerda (BELYADI e HAGHIGHAT, 2021).

Uma parâmetro importante dentro deste processo é o tamanho dos passos tomados dentro do gradiente descendente, também chamado de taxa de

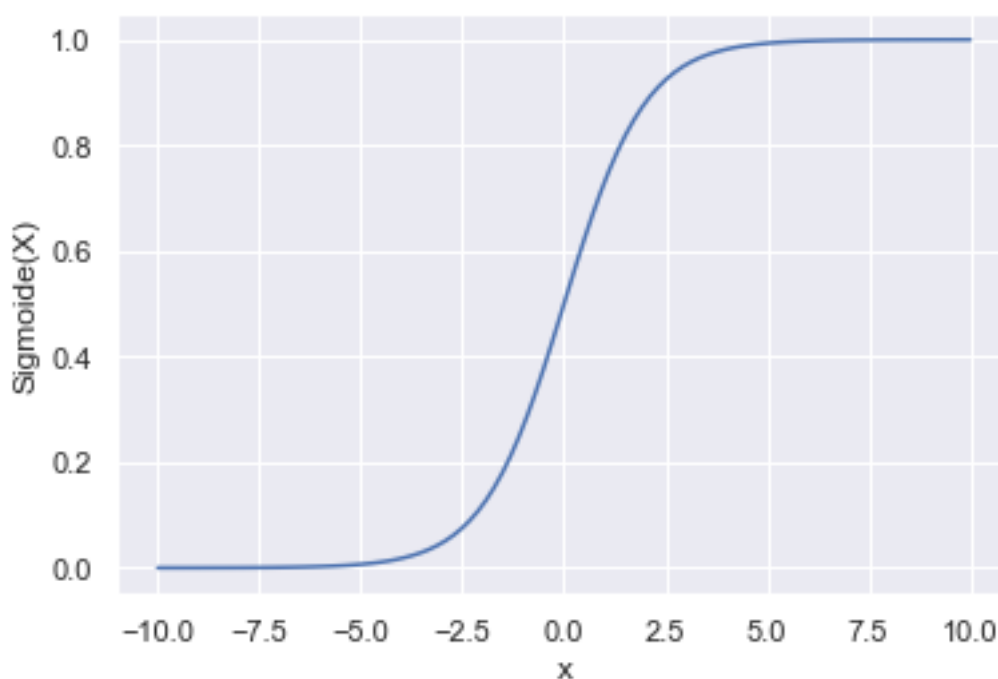
aprendizagem. Se esta taxa for muito pequena, o algoritmo passa por muitas iterações, tornando o processo de aprendizado lento. Por outro lado, se o valor da taxa for muito grande, o processo poderá pular o valor mínimo e acabar do outro lado da curva. Isso poderá fazer o algoritmo divergir, alcançando valores cada vez maiores e falhando em encontrar uma boa solução (GÉRON, 2019).

2.3.5 FUNÇÕES DE ATIVAÇÃO

Como mencionado anteriormente, a operação básica de um neurônio artificial envolve a soma dos sinais de entrada ponderados e a aplicação de uma função de ativação. Na camada de entrada, essa função normalmente é do tipo identidade. Porém, para alcançar as vantagens das redes com múltiplas camadas, é necessária a utilização de redes não lineares (FAUSETT, 1994).

A função sigmoide, apresentada na Figura 6, possui um gráfico em forma de S e é a forma mais comum de função de ativação utilizada na construção de redes neural. Essa função é definida de forma estritamente crescente e exibe um equilíbrio entre comportamentos lineares e não lineares (HAYKIN, 1999).

Figura 6 - Gráfico da função sigmoide



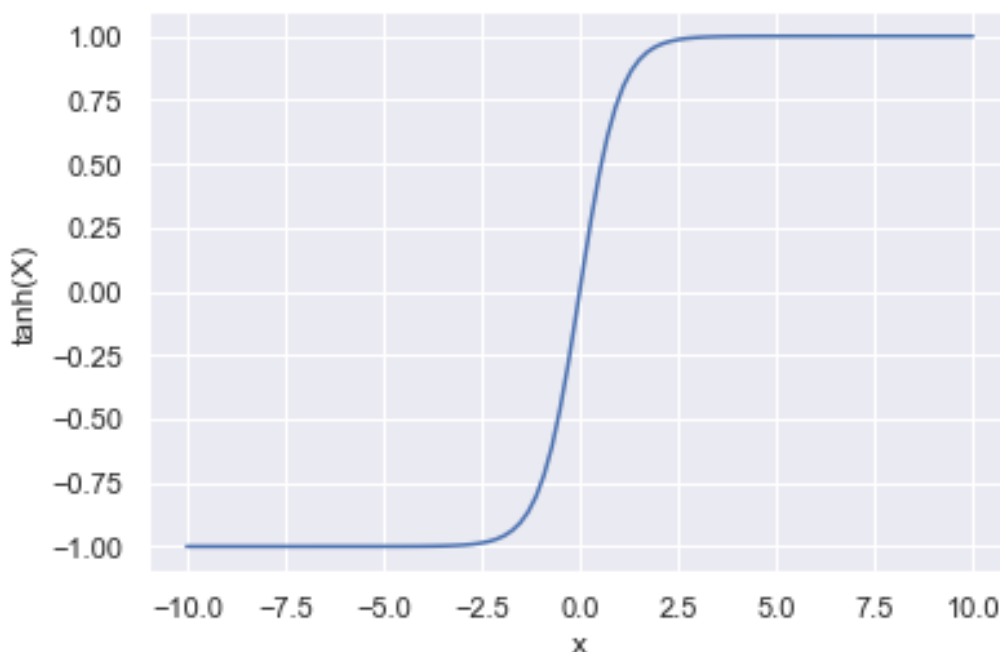
Fonte: Autoria própria (2021)

A função sigmoide, definida na Equação 2, transforma as entradas de uma rede neural artificial em uma saída com um valor dentro do intervalo (0, 1). Por essa razão, essa função também é chamada de função de achatamento: essa função achata qualquer valor no intervalo $(-\infty, \infty)$ para um valor no intervalo (0, 1) (ZHANG, *et al.*, 2021).

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \quad (2)$$

A função da tangente hiperbólica (tanh), apresentada na Figura 7, é similar à função sigmoide com a diferença de que seu intervalo dos valores de saída está entre (-1, 1). A função tanh é preferível a função sigmoide, uma vez que ela é centrada em zero e possui um desempenho de gradiente melhor (BELYADI e HAGHIGHAT, 2021)

Figura 7 - Gráfico da função tanh



Fonte: Autoria própria (2021)

Assim como a função sigmoide, a função tanh também achata os valores de entrada, transformando em elementos no intervalo entre -1 e 1. A função tanh, definida na Equação 4, se aproxima de uma transformação linear quando os valores de entrada estão próximos de zero e, apesar de sua similaridade com a função

sigmoide, a função tanh apresenta simetria de ponto em relação à origem do sistema de coordenadas (ZHANG *et al.*, 2021).

$$\tanh(x) = \frac{1 - \exp(-2x)}{1 + \exp(-2x)} \quad (4)$$

2.3.6 OTIMIZAÇÃO DE HIPERPARÂMETROS

De maneira geral, a construção de um modelo de aprendizado de máquina é um processo complexo e demorado que envolve a determinação de um algoritmo de aprendizagem apropriado e a obtenção de uma arquitetura otimizada por meio da obtenção de um conjunto de hiperparâmetros. Esse tipo de parâmetro não pode ser estimado durante o processo de aprendizagem, como os pesos da rede, e devem ser definidos antes do treinamento do modelo, pois são eles que definem a arquitetura do modelo (YANG; SHAMI, 2020).

Wu *et al.*(2019) afirmam que existem dois tipos principais de otimização de hiperparâmetros. A primeira é a busca manual, onde os parâmetros são definidos de forma manual, dependendo da intuição de usuários experientes que conseguem identificar parâmetros que impactam de forma positiva nos resultados. Já na busca automática, algoritmos realizam uma busca em diversas combinações de parâmetros e avaliam os resultados de acordo com uma métrica pré-definida.

Porém, esse tipo de busca possui um custo computacional e de tempo muito alto. Para tentar resolver esse tipo de problema, outros algoritmos podem ser utilizados, como por exemplo a busca bayesiana.

Essa técnica utiliza um modelo preditivo chamado de “substituto” para modelar o espaço de busca e alcançar uma boa combinação de valores dos parâmetros o mais rápido possível, já que uma busca exaustiva em todas as combinações se tornaria muito complexa em um espaço de pesquisa com inúmeros parâmetros (SHCHERBATYI *et al.*, 2020).

2.4 AVALIAÇÃO ESTATÍSTICA

O aumento na utilização de modelos de predição na tentativa de resolução dos mais diversos tipos de problemas aumentou também o interesse em determinar quais formulações produzem resultados mais precisos. Por sua vez, a utilização de medidas de erro estatísticas passaram a ser amplamente utilizadas (WILLMOTT; MATSUURA, 2005).

Uma medida estatística amplamente utilizada na avaliação de modelos de regressão multivariada é o Erro Médio Absoluto (MAE). Outra forma de avaliar o erro médio é por meio da Raiz do Erro Quadrático Médio (RMSE), porém o MAE supera os resultados alcançados pela RMSE na avaliação da precisão de modelos em quase todas as situações. A exceção ocorre quando o erro esperado satisfaz uma distribuição gaussiana e amostra de treinamento suficientes estão disponíveis (QI *et al.*, 2020).

O MAE, descrito na Equação 5, é uma métrica do tipo diferença absoluta. Esse tipo de métrica limita o impacto de pontos discrepantes no desempenho de modelos e é, portanto, particularmente eficiente em situações onde existem problemas de qualidade no conjunto de dados (STEURER; HILL; PFEIFER, 2020)

$$\frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (5)$$

Nessa métrica, o n representa o número de observações no conjunto de dados, y_j representa o valor real observado e o \hat{y}_j representa o valor alcançado pelo modelo utilizado.

Uma alternativa para as métricas de diferença absoluta são os erros quadráticos médios. Esse tipo de técnica é mais sensível a pontos discrepantes e possui utilidade em situações onde erros grandes de predição necessitam ser minimizados (STEURER; HILL; PFEIFER, 2020).

A primeira dessas técnicas é o MSE, apresentado na Equação 6. Nessa métrica o n corresponde ao tamanho do conjunto de dados, o y_j caracteriza o valor real observado e o \hat{y}_j equivale ao valor de predição do modelo para cada observação.

$$\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2 \quad (6)$$

Outra técnica do tipo erro médio quadrático é a RMSE, apresentada na Equação 7. Steurer, Hill e Pfeifer (2020) dão preferência a essa métrica devido a sua facilidade de interpretação por parte do usuário, causada pelos valores menores gerados por este tipo de erro.

$$\sqrt{\frac{\sum_{j=1}^n (y_j - \hat{y}_j)^2}{n}} \quad (7)$$

Nesse tipo de erro, o n configura o tamanho total do conjunto de dados, o y_j corresponde ao valor real de cada observação e o \hat{y}_j configura o valor alcançado pelo modelo de predição.

2.4.1 NORMALIZAÇÃO

Para garantir que um algoritmo de aprendizado não seja influenciado pela magnitude dos dados, estes devem ser primeiramente escalonados. Este processo também pode acelerar os algoritmos de otimização que serão utilizados no desenvolvimento do modelo, tendo em vista que cada um dos valores de entrada estarão aproximadamente no mesmo intervalo (BELYADI e HAGHIGHAT, 2021).

A normalização dos dados é um passo de pré-processamento importante que envolve a transformação das características em um intervalo comum, onde os valores maiores não dominem as características com uma faixa de valor menor. Este processo é útil em métodos de aprendizado estatístico já que todas as características contribuem igualmente para o processo de aprendizagem (SINGH; SINGH, 2020).

A normalização dos dados garante que cada uma das características estará em uma escala de 0 a 1, diminuindo assim o peso que o modelo pode dar para aspectos dos dados em diferentes magnitudes. Uma das desvantagens dessa técnica é que a limitação dos valores em uma escada de 0 a 1 gera um desvio

padrão muito pequeno, podendo esconder assim os efeitos que valores discrepantes possuem no conjunto de dados (BELYADI; HAGHIGHAT, 2021).

2.4.2 GENERALIZAÇÃO DE MODELOS

Um conceito importante na área de aprendizado de máquina é o chamado *bias-variance tradeoff*. O viés é definido como a simplificação da hipótese do modelo, ou a sua incapacidade de entender os verdadeiros relacionamentos entre as características de entrada e saída. Já a variância se refere a variabilidade das previsões do modelo para determinado ponto dos dados. Uma variância alta indica uma falta de capacidade em prever novos dados devido a uma baixa generalização (BELYADI e HAGHIGHAT, 2021).

Um modelo desenvolvido com uma boa generalização irá produzir um mapeamento correto entre as entradas e saídas, mesmo quando os dados de entrada são diferentes dos exemplos utilizados durante o treinamento. Porém, um modelo pode acabar memorizando os dados utilizados nesse processo, dando uma importância muito grande para características que existem apenas nos dados de treinamento, mas que não são relevantes para a função a ser modelada. Esse fenômeno é conhecido como *overfitting* ou *overtraining* (HAYKIN, 1999).

A utilização da técnica de aprendizagem por *backpropagation* deve garantir a criação de uma rede neural bem treinada e que tenha aprendido o suficiente sobre o passado para generalizar o futuro. Com base nessa perspectiva, o processo de aprendizado se torna uma escolha de parametrização da rede neural de acordo com um conjunto de dados. Nesse contexto, a ferramenta estatística de validação cruzada pode ser utilizada para realizar essa seleção e desenvolver uma rede com boa generalização (HAYKIN, 1999).

Na validação cruzada do tipo *k-fold*, os dados são divididos em k subconjuntos do mesmo tamanho. Após isso, o processo de aprendizado é executado k vezes. Em cada uma dessas repetições, $1/k$ dos dados é utilizado como o conjunto de testes e os exemplos restantes são utilizados no treinamento do modelo. Valores populares para k estão entre cinco e dez, o suficiente para alcançar uma estimativa que é estatisticamente provável de ser precisa (RUSSELL; NORVIG, 1995).

2.4.3 TÉCNICAS DE DEEP LEARNING

Apesar de ter sido aplicada em diversos problemas com sucesso, as redes neurais possuem limitações importantes, como a falta de garantia de convergência ao erro mínimo global pelo algoritmo *backpropagation*. Nessa implementação, o gradiente descendente pode acabar ficando preso em um dos múltiplos mínimos locais ao tentar diminuir o erro entre os valores alvos nos exemplos de treinamento e as saídas da rede (MITCHELL, 1997).

Apesar de ser uma ferramenta de aprendizagem de máquina capaz de abordar problemas complexos nas mais diversas áreas, as redes neurais artificiais possuem algumas desvantagens. Nessa técnica, a duração da fase de aprendizado pode ser demorada para grandes conjuntos de dados e esse aprendizado pode acabar resultando na obtenção de diversos mínimos locais (KUMAR et al., 2016).

Os algoritmos de redes neurais clássicas são ineficientes no processamento de grandes conjuntos de dados devido a sua dificuldade em melhorar a qualidade das previsões realizadas e limitações em escalabilidade e capacidades de generalização. O rápido desenvolvimento da inteligência artificial trouxe a proposição de uma nova ferramenta conhecida como Deep Learning (CAI et al., 2019).

Deep Learning é um ramo do aprendizado de máquina, composto por uma série de algoritmos que tentam modelar abstrações de alto nível nos dados por meio da utilização de um grafo profundo, com diversas camadas de processamento contendo transformações lineares e não lineares (BARBEDO, 2018).

Esse método utiliza uma estrutura de múltiplas camadas ocultas para melhorar a capacidade de classificação e ajuste de grandes conjuntos de informações e de dados com múltiplas características. Comparado com redes neurais clássicas, esse método demonstra um poder de computação robusto e tem sido aplicado com sucesso em reconhecimento de imagens, motores de busca, previsão do preço de ações e outros campos (CAI et al., 2019)

Deep Learning representa um grande avanço no campo do aprendizado de máquina. A capacidade de aprendizado em múltiplos níveis e sem a necessidade de características projetadas manualmente permitiu que esse tipo de aprendizagem fosse capaz de alcançar avanços importantes em diversas áreas complexas da inteligência artificial (FERREIRA et al., 2019).

3 MATERIAIS E MÉTODOS

Devido à natureza do problema estudado e a forma como este trabalho se propôs a resolvê-lo, esta pesquisa pode ser definida, em relação a abordagem utilizada, como do tipo quantitativa. Haddadi et al. (2017) definem que esse tipo de pesquisa tenda a ser relacionada ao positivismo e a busca pela coleta de dados corretos, com o objetivo de estudar relações entre fatos e se esses relacionamentos estão de acordo com teorias e resultados anteriores.

Estas características se encontram nessa pesquisa através da utilização de procedimentos estruturados na coleta de dados, quantificação dos resultados alcançados e a utilização de métodos estatísticos na análise dos dados.

Já em relação a natureza do trabalho desenvolvido, esta pesquisa pode ser definida como sendo do tipo aplicada. Segundo Gerhardt e Silveira (2009) esse tipo de pesquisa busca gerar conhecimentos que terão aplicação prática na solução de problemas específicos. Espera-se que os resultados alcançados pela execução do processo de criação de modelos de RNAs aplicadas à predição da precipitação da água de microaspersores possa ser aplicado de maneira prática na solução de problemas específicos no campo da agricultura irrigada.

Esta pesquisa busca entender a razão da ocorrência de determinados fenômenos, como, qual conjunto de variáveis possui um peso maior na predição em sistemas de irrigação. Neste contexto, este trabalho pode ser definido, em relação aos seus objetivos, como sendo do tipo explicativo. Segundo Razavian, Paech e Tang (2019), esse tipo de investigação é mais profundo que a pesquisa descritiva já que ele busca descrever fenômenos e tenta explicar a razão pela qual determinados comportamentos acontecem e por que eles acontecem dessa forma.

Essa pesquisa tem como objetivo encontrar o melhor conjunto de hiperparâmetros a serem utilizados em uma arquitetura de um modelo de rede neural artificial que seja capaz de realizar a predição do comportamento de sistemas de irrigação por microaspersão e melhorar o desempenho dos mesmos.

Além disso, este trabalho também se propôs a construir um sistema *web* que irá utilizar os resultados dos modelos desenvolvidos no apoio a tomada de decisão por partes dos gestores responsáveis pela modelagem de sistemas de irrigação otimizados. Estas características tornam este trabalho, portanto, uma pesquisa do

tipo operacional. Esse tipo de pesquisa pode ser definido, segundo Bargos, Lamas e Bilato (2018), como um conjunto de métodos e técnicas aplicadas com o objetivo de alcançar soluções ótimas ou quase ideais para problemas de tomada de decisão complexas.

3.1 AMBIENTE DE DESENVOLVIMENTO

Para o desenvolvimento deste projeto foi utilizado o conjunto de ferramentas Anaconda¹. Esta plataforma de distribuição da linguagem de programação Python utilizada no desenvolvimento de soluções de aprendizado de máquina e ciência de dados.

Python² é uma linguagem de programação orientada a objetos, interpretada e interativa. Criada por Guido van Rossum em 1991, atualmente é gerenciada pela organização sem fins lucrativos Python Software Foundation.

O ambiente de desenvolvimento utilizado para realizar a implementação dos modelos, a seleção de seus hiperparâmetros e a avaliação dos resultado alcançados foi o Jupyter Notebook, uma aplicação *web* interativa para o desenvolvimento de código em Python.

3.2 PROCESSO DE DESENVOLVIMENTO

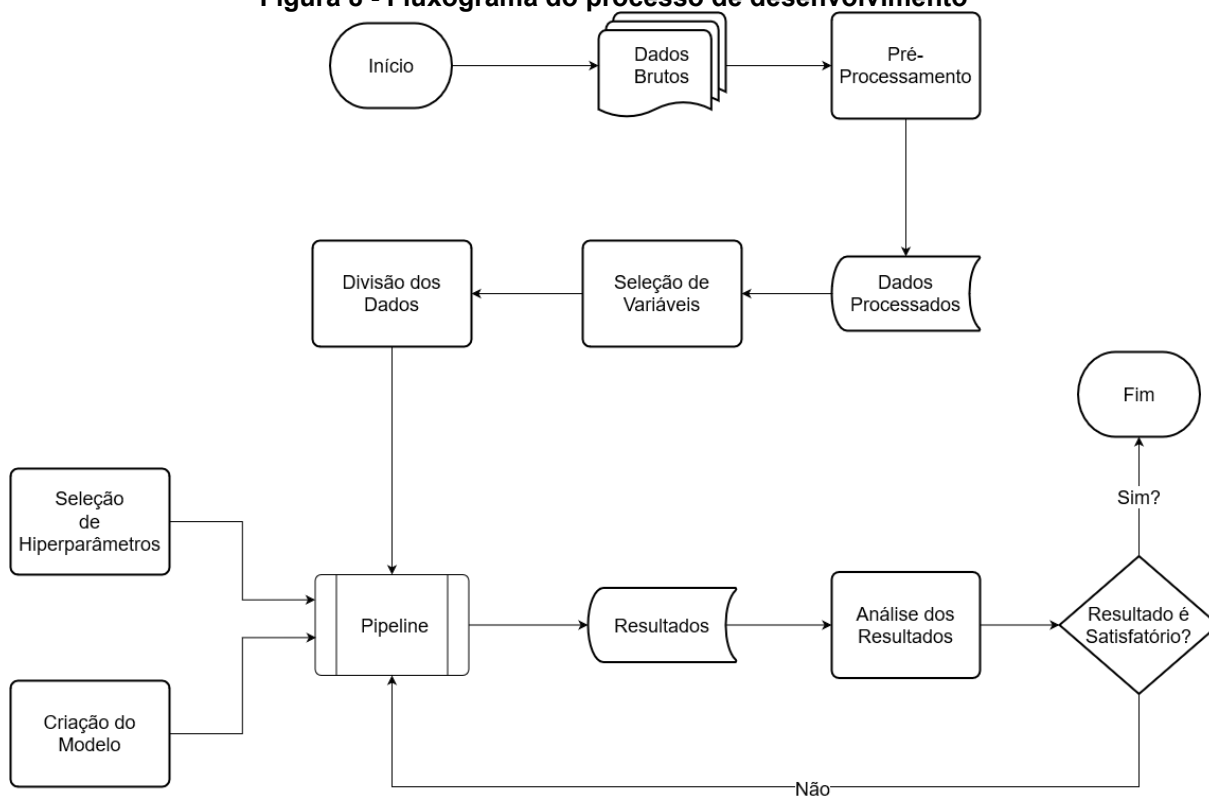
O processo de execução da pesquisa desenvolvida neste trabalho, apresentado na Figura 8, pode ser dividido em três grandes etapas:

- Pré-processamento dos dados;
- Criação de modelos e seleção de hiperparâmetros;
- Execução do processo de *pipeline* e análise dos resultados.

¹ <https://www.anaconda.com/>

² <https://www.python.org/>

Figura 8 - Fluxograma do processo de desenvolvimento



Fonte: Autoria própria (2021)

O processo de desenvolvimento se iniciou com a transformação dos dados brutos em um formato que facilita a leitura, manipulação, análise das informações. Como resultado deste pré-processamento foi gerado um arquivo com os dados processados a ser utilizado como entrada dos modelos de previsão a serem desenvolvidos. Após isso, a seleção das variáveis independentes utilizadas para realizar a estimativa da precipitação foi realizada e, por fim, os dados a serem carregados nos modelos desenvolvidos foram divididos em conjuntos de dados de treinamento e teste.

Após essa etapa foi realizada a criação do modelo de previsão, através da definição da arquitetura e escolha de características como número de número de camadas, função de custo e métricas de avaliação. Outro processo realizado nesta etapa foi a seleção dos hiperparâmetros da arquitetura do modelo. Estes hiperparâmetros foram avaliados em etapas posteriores de desenvolvimento, com o objetivo de encontrar a melhor arquitetura para um modelo, baseado em redes neurais, a ser utilizado na previsão da quantidade de água utilizada por microaspersores.

Por fim, os dados pré-processados foram utilizados em conjunto com os modelos desenvolvidos e os hiperparâmetros selecionados na execução de um processo de *pipeline*. Essa técnica realiza uma busca, seja ela aleatória ou exaustiva, pela combinação de hiperparâmetros que alcançará os melhores resultados na resolução do problema proposto.

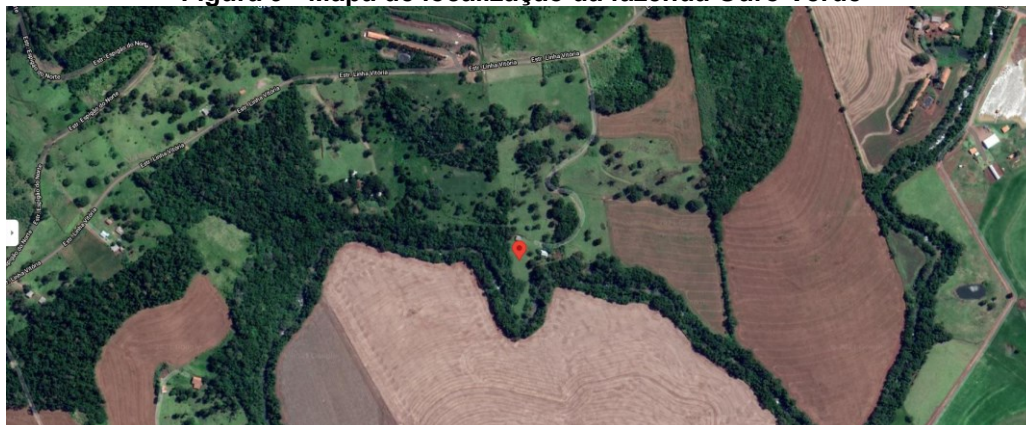
Os resultados alcançados pelo processo de *pipeline* foram gravados em um arquivo e uma análise dos resultados foi então executada. Essa avaliação estatística foi realizada para garantir que o modelo gerado pelo processo de *pipeline* é, de fato, o que alcançará o melhor resultado quando aplicado a predição da precipitação de microaspersores.

O processo de *pipeline* continuou sendo executado enquanto o modelo de predição não produzisse resultados satisfatórios, definidos de acordo com métricas de erro e generalização pré-definidas.

3.3 CONJUNTO DE DADOS

Em Menezes et al. (2015) redes neurais artificiais foram desenvolvidas para simular a precipitação de microaspersores utilizando valores de pressão, bocal e dados meteorológicos. Os dados desta pesquisa foram coletados na fazenda Ouro Verde, apresentada na Figura 9, está localizada no município de Medianeira, Brasil ($25^{\circ}12'19,26''S; 54^{\circ}3'27,93''W; 360m$). Duzentos e cinquenta e seis pluviômetros, distribuídos em uma matriz 16×16 , foram utilizados para medir a quantidade de água utilizada.

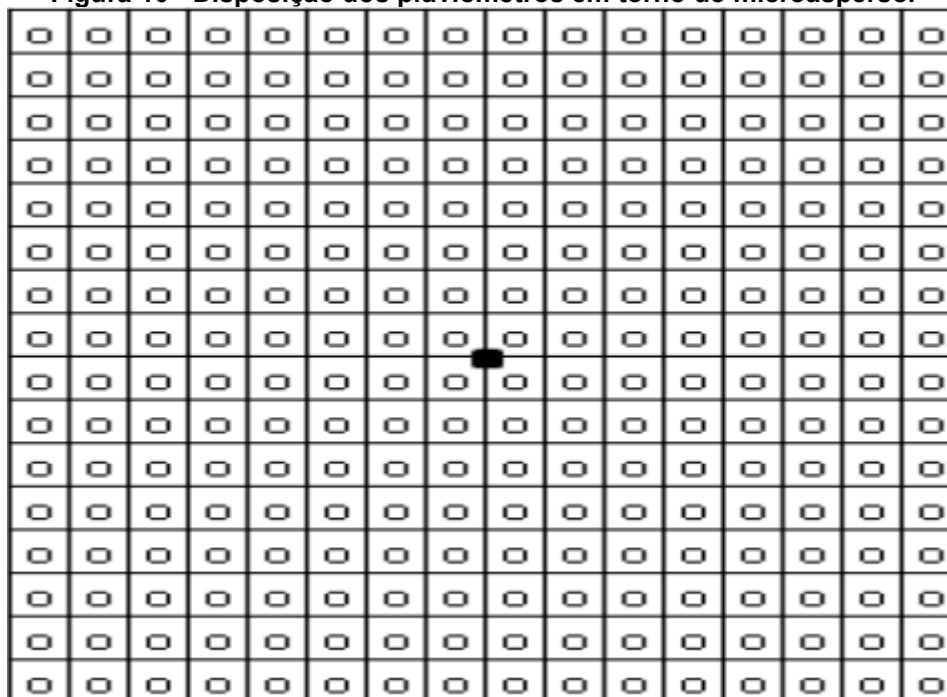
Figura 9 - Mapa de localização da fazenda Ouro Verde



Fonte: Autoria própria (2021)

Nesta configuração, apresentada na Figura 10, os pluviômetros se encontram dispostos em uma malha, com um microaspersor no centro, posicionados a uma distância de 1,5m um do outro e uma altura de 1,25m em relação à superfície.

Figura 10 - Disposição dos pluviômetros em torno do microaspersor



Fonte: Adaptado de Bernardo (2009)

Nove configurações de características de diâmetro do bocal (mm) e quebra jato do microaspersor Pingo giro completo 360° da marca Fabrimar foram avaliadas. Os valores testados foram: 2.6 x 2.4, 2.8 x 2.4, 3.0 x 2.6, 3.2 x 2.6, 3.4 x 2.6, 3.6 x 2.8, 3.8 x 2.8, 4.0 x 2.8 e 4.0 x 3.2. Todas as configurações foram testadas em quatro valores de pressão (kgf/cm³) diferentes: 2.0, 2.5, 3.0 e 3.5.

Dados meteorológicos, como velocidade média (m/s), direção do vento (graus), valores de vazão do microaspersor e data e horário de execução dos experimentos também foram coletados. Foram realizadas três repetições para cada uma das configurações, totalizando 27 ensaios com duração de 2 horas cada.

3.4 PRÉ-PROCESSAMENTO E ANÁLISE DOS DADOS

Os dados apresentados em Menezes et al. (2015) são formados por 27 arquivos no formato Excel. Cada um destes arquivos possui quatro planilhas contendo os dados coletados para cada uma das diferentes do microaspersor.

Devido a esta formatação, um pré-processamento inicial foi realizado para transformar os dados destes arquivos em um formato que facilite sua manipulação pelos modelos desenvolvidos.

Esse processo foi realizado em todas as planilhas existentes nos 27 arquivos Excel por meio da utilização do pandas, um pacote de classes e métodos baseado na biblioteca NumPy. Conjuntos de dados podem vir das mais diversas fontes e nos mais variados formatos, como coleções de imagens, áudios ou medidas numéricas. Por essa razão, o armazenamento e manipulação destes dados é muito importante. O NumPy (abreviatura de *Numerical Python*) fornece uma interface eficaz para armazenar e operar *buffers* de dados. Vetores NumPy fornecem operações para manipulação e armazenamento dos dados de forma mais eficiente, especialmente quando estes vetores aumentam em tamanho (VANDERPLAS, 2017).

Baseado no NumPy, o pandas é um pacote que tem como destaque sua implementação do objeto DataFrame, que é essencialmente um vetor multidimensional, com rótulos para linhas e colunas, e frequentemente estes objetos possuem tipos de dados heterogêneos ou mesmo faltantes. Além de oferecer uma interface conveniente para armazenar dados rotulados, o pandas implementa uma série de operações de manipulação de dados familiar, tanto para usuários de sistemas gerenciadores de bancos de dados quanto para usuários de programas de planilhas (VANDERPLAS, 2017).

3.5 SELEÇÃO DE VARIÁVEIS E DIVISÃO DOS DADOS

Na execução desta pesquisa foram utilizadas duas configurações do conjunto de dados em relação às variáveis independentes a serem utilizadas pelos modelos de estimativa construídos. A primeira configuração utiliza as seguintes variáveis: bocal (mm), pressão (kgf/cm^3), quebra-jato, velocidade média (m/s) e direção do vento (graus).

A segunda configuração utiliza, além das variáveis descritas na primeira configuração e as variáveis horário do experimento, dia do experimento, vazão inicial, final e total do microaspersor.

Esta divisão entre duas configurações de variáveis foi utilizada para verificar se a utilização de outras variáveis, em conjunto com os principais dados do microaspersor e dados meteorológicos, impactam de alguma maneira na qualidade das previsões realizadas pelos modelos desenvolvidos.

Em ambas as configurações de variáveis, o conjunto de dados foi dividido da seguinte maneira: por volta de 7% dos dados foram separados, de forma aleatória, como conjunto de validação dos resultados do melhor modelo encontrado. Foram criadas duas configurações para a divisão dos dados restantes. A primeira configuração foi dividida em 70% para o conjunto de treinamento e 30% para o conjunto de teste. E na segunda configuração, os dados foram divididos em 80% para o treinamento e 20% para o teste. A criação de duas configurações para a divisão dos dados foi realizada para verificar se os resultados alcançados pelos modelos desenvolvidos serão impactados pelo tamanho escolhido para os conjuntos de treinamento e teste.

O processo de divisão dos dados foi realizado através do método `train_test_split` do `scikit-learn`³, uma biblioteca escrita em Python utilizada no desenvolvimento de algoritmos de aprendizado de máquina e para o pré-processamento de dados.

O método `train_test_split` permite a divisão aleatória de um conjunto de dados em duas partes, definindo a porcentagem dos dados que formarão o conjunto de treinamento e designando o restante como o conjunto de teste.

3.6 DESENVOLVIMENTO DOS MODELOS

A biblioteca Keras⁴ foi utilizada na criação dos modelos de predição desenvolvidos nesta pesquisa. O Keras é uma API de Deep Learning, escrita em Python. Essa ferramenta é uma camada superior da plataforma de aprendizado de máquina TensorFlow⁵, e ela foi desenvolvida com foco em possibilitar rápida experimentação (CHOLLET et al., 2015).

³ <https://scikit-learn.org/>

⁴ <https://keras.io/>

⁵ <https://www.tensorflow.org/>

O TensorFlow é uma plataforma de aprendizado de máquina de código aberto e ponta-a-ponta. Essa camada de infraestrutura para programação diferenciável possui capacidades como execução eficiente de operações de baixo nível, computação escalonável nos mais diversos dispositivos e a capacidade de exportar programas para servidores ou *navegadores* externos (CHOLLET et al., 2015).

Brownlee (2017) descreve o Keras como uma biblioteca minimalista em Python, criada com o objetivo de desenvolver modelos de Deep Learning tanto para pesquisa quanto para desenvolvimento. Esta biblioteca foi desenvolvida e é mantida seguindo quatro princípios orientadores:

- Modularidade: um modelo pode ser entendido como uma sequência ou grafo;
- Minimalismo: a biblioteca fornece o suficiente para alcançar um objetivo, maximizando a legibilidade;
- Extensibilidade: novos componentes são fáceis de adicionar e usar dentro da estrutura, com a intenção de facilitar a exploração de novas ideias.

Nesta pesquisa foram criadas duas arquiteturas de modelos utilizando a biblioteca Keras: o primeiro modelo possui apenas uma camada oculta e um modelo possuindo entre três e seis camadas ocultas. Ambos os modelos testados possuem uma camada de entrada de tamanho 5 ou 10 neurônios, dependendo do número de variáveis sendo avaliadas e uma camada de saída com 256 neurônios, que é o número de pluviômetros utilizados na coleta de dados.

Esta estratégia foi utilizada para verificar se existe alguma diferença nos resultados alcançados com a utilização de um modelo de RNA raso em comparação com os observados em uma arquitetura profunda.

Estes modelos foram desenvolvidos utilizando o modelo Sequential do Keras. Esta é uma implementação direta de uma simples lista de camadas contendo uma única entrada e uma única saída. Por meio da utilização do método `add`, que adiciona uma nova instância no topo da pilha, novas camadas de variados tipos são inseridas em ambos os modelos construídos nessa pesquisa.

O primeiro tipo de camada adicionado ao modelo Sequencial é uma instância da classe Dense, uma camada de RNA totalmente conectada padrão. O modelo raso possui apenas uma dessas camadas entre a camada de entrada e a camada de saída e o modelo profundo possui de três a seis camadas do tipo Dense.

Após a inserção de uma camada Dense no modelo, uma camada do tipo Dropout também foi adicionada. Uma camada Dropout define, de forma aleatória, o valor das unidades de entrada para zero durante a fase de treinamento. Este processo é realizado para ajudar a prefinir *overfitting*. Uma camada do tipo Dropout foi adicionada apenas uma vez o modelo raso, enquanto o modelo profundo possui de três a seis camadas Dropout.

3.7 SELEÇÃO DE HIPERPARÂMETROS

No modelo raso, o primeiro argumento explorado foi o número de neurônios da camada a ser adicionada ao modelo. Os valores analisados foram: 128 neurônios, 256 neurônios, 512 neurônios, 768 neurônios e 1024 neurônios. O próximo argumento avaliado foi a função de ativação utilizada pelo neurônio. Se nenhuma função for especificada, uma ativação linear é aplicada. As funções testadas como hiperparâmetros para este argumento foram as seguintes: GeLu, ReLu, Swish, Mish, Sigmoid, Softplus, Softsign, Tanh.

Na sequência, a forma de inicialização da matriz de pesos da camada adicionada ao modelo foi avaliada como um hiperparâmetro. Os tipos verificados foram: Uniforme, Normal, Glorot_Normal e Glorot_Uniform. Outro hiperparâmetro testado foi o tipo de função de restrição que deverá ser aplicada à matriz de pesos. O tipo escolhido para este argumento foi a MaxNorm, que limita os pesos incidentes a cada um dos neurônios da camada oculta a terem uma norma matricial menor ou igual a um valor desejado. Os valores verificados para este argumento foram: um, dois, três, quatro e cinco.

O tipo de função regularizadora aplicada à matriz de pesos também foi examinado como um hiperparâmetro. Os valores testados foram: função de regularização L1, função de regularização L2 e função de regularização L1_L2. Todas as funções listadas foram avaliadas com seus fatores de regularização no valor de 0,0001. Outro tipo de hiperparâmetro avaliado foi a taxa de abandono, que

determinada a fração de unidades de entrada a serem definidas com o valor zero. Entre os valores testados estão: 0.0, 0.1, 0.2, 0.3, 0.4 e 0.5.

Os argumentos que definem a configuração do modelo a ser treinado também foram avaliados. O primeiro parâmetro investigado foi o algoritmo de aprendizado a ser utilizado. Os valores verificados para este argumento foram: Adam, Gradiente Descendente Estocástico, RMSprop e Adagrad. A taxa de aprendizado de cada um destes algoritmos também foi testada como um hiperparâmetro. Os valores verificados foram: 10^{-1} , 10^{-2} , 10^{-3} , 10^{-4} , 10^{-5} e 10^{-6} .

Por fim, outros parâmetros não relacionados diretamente a construção do modelo de predição também foram analisados. Entre estes parâmetros estão: número de épocas, o tamanho do batch de treinamento e o método de normalização dos dados de entrada.

Os números de épocas testados foram: 400, 500, 600 e 700. Já os valores para o tamanho do mini-lote de treinamento avaliados foram: 6, 9 e 18. Por fim, os métodos de normalização utilizados como hiperparâmetros foram: StandardScaler, MinMaxScaler, RobustScaler, Power Transformer e QuantileTransformer dos tipos uniform e normal. A implementação destes métodos foi realizada com o auxílio da biblioteca scikit-learn.

Todos os hiperparâmetros descritos nesta seção foram testados no modelo raso com apenas uma camada. Para o modelo profundo foram verificados apenas três parâmetros: número de épocas, tamanho do mini-lote de treinamento e número de camadas. Os valores verificados para o número de épocas foram: 100, 200, 300 e 400. Já para o mini-lote de treinamento foram analisados os valores de 6, 9 e 18. Por fim, foram verificados quatro valores para o número de camadas: três camadas, quatro camadas, cinco camadas e seis camadas.

Para os hiperparâmetros restantes, foram utilizados os valores que a busca realizada para o modelo raso determinou como sendo o conjunto de hiperparâmetros capaz de realizar a predição da água utilizada por microaspersores de forma mais precisa.

A busca pelo melhor conjunto de hiperparâmetros para os modelos desenvolvidos neste projeto foi feita com o auxílio do scikit-optimize⁶. Esta biblioteca,

⁶ <https://scikit-optimize.github.io>

baseada no scikit-learn, que utiliza uma implementação do algoritmo de busca bayesiana para encontrar os parâmetros que alcançam os melhores resultados.

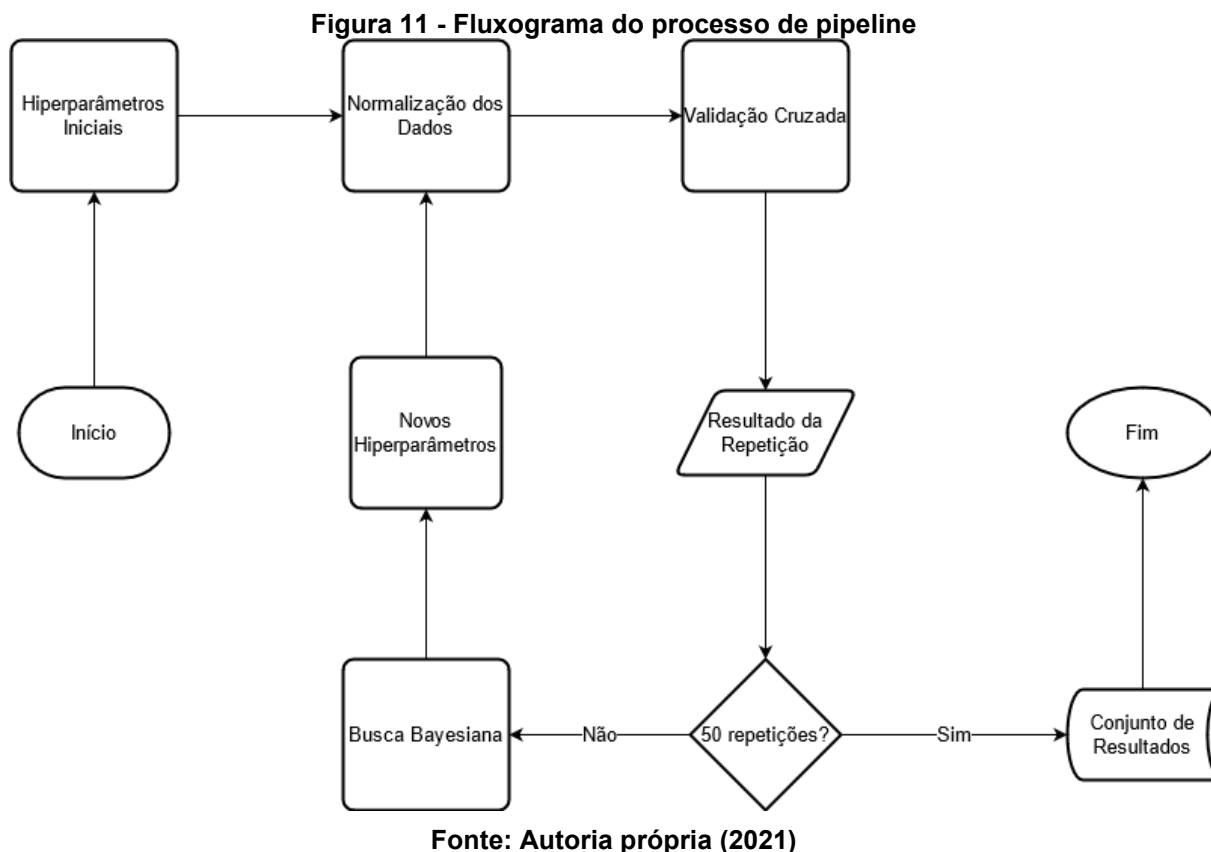
A configuração do processo de busca executado pela biblioteca scikit-optimize possui 50 iterações, além de executar uma validação cruzada de 10 divisões para cada repetição. No modelo raso, este procedimento foi executado em cada uma das configurações de variáveis até que um resultado satisfatório foi encontrado. No modelo profundo, o número menor de hiperparâmetros testados permite a realização de uma busca completa para encontrar a melhor configuração. Esta situação permite a realização de uma busca única para todas as configurações de variáveis e divisão dos dados, adicionando mais quatro experimentos ao processo.

3.8 PIPELINE

Um passo importante na criação dos modelos de estimação da precipitação de água de microaspersores é a normalização dos dados. Como foi descrito na Seção 3.7, essa característica foi avaliada como um hiperparâmetro para que possa ser encontrado o método de transformação de dados que garanta uma predição superior aos demais métodos testados.

Porém, nos modelos desenvolvidos com a ajuda do Keras não existe um parâmetro específico para informar o tipo de transformação e aplicá-lo nos dados antes da busca de hiperparâmetros seria uma tarefa muito custosa, pois ela teria que ser repetida para cada um dos algoritmos de transformação.

Para resolver este problema, um processo de *pipeline* foi desenvolvido. Este procedimento, descrito na Figura 11, foi executado com a ajuda do método *Pipeline* da biblioteca scikit-learn. Este método é utilizado para executar uma série de passos sequenciais e a aplicação de um processo de validação cruzada.



A *pipeline* desenvolvida neste trabalho possui dois passos. O primeiro é o *scaler*, ou transformador e por último vem o *estimator*, ou modelo. Como o método *pipeline* aplica os passos em sequência, a primeira etapa que a busca de hiperparâmetros irá realizar será encontrar um método de transformação e aplicá-los aos dados.

Somente após isso, os demais argumentos foram aplicados e uma validação cruzada, utilizando os dados transformados e o modelo criado, foi executada. Para finalizar uma etapa deste processo, as métricas alcançadas foram salvas para uma análise posterior. Como definido anteriormente, 50 repetições desse processo foram realizadas.

3.9 ANÁLISE ESTATÍSTICA DOS RESULTADOS

Técnicas estatísticas foram utilizadas para verificar a eficácia de predição dos modelos desenvolvidos, a qualidade das predições realizadas e a capacidade de generalização. Para verificar a qualidade das predições realizadas foram utilizadas as técnicas estatísticas Erro Médio Absoluto (MAE), Erro Quadrático Médio

(MSE) e a Raiz do Erro Quadrático Médio (RMSE). A verificação do ajuste do modelo e de sua capacidade de generalização foi feita através do Coeficiente de Determinação (R^2) e de gráficos de curvas de *overfitting/underfitting*.

Por fim, testes de normalização foram executados nos resíduos dos modelos desenvolvidos. Os testes realizados foram: Shapiro-Wilk, D'Agostino Pearson, Lilliefors e Anderson-Darling.

As técnicas estatísticas e testes de normalização citadas nesta seção foram desenvolvidas com a ajuda da biblioteca scikit-learn, que disponibiliza implementações de todos os métodos utilizados.

3.10 AMBIENTE *WEB*

Por fim, um ambiente *web* foi desenvolvido para apresentar com melhor clareza os resultados obtidos com os modelos com a maior capacidade de predição. Esta aplicação foi criada com a ajuda da biblioteca Plotly Dash, um *framework* escrito em Python e utilizado na construção de aplicações *web* analíticas.

O aplicativo desenvolvido possui uma página, na qual é possível escolher um dos experimentos do conjunto de validação e um dos modelos desenvolvidos. Esse modelo realiza uma predição com base no experimento escolhido. A página apresenta a comparação da predição do modelo em relação aos valores reais por meio de uma tabela contendo os valores de precipitação, um gráfico de superfície simulando o formato da precipitação da água, um gráfico de linhas do perfil de distribuição e por fim uma tabela contendo os valores do CUC para as distâncias, em metros, de 12x12, 12x15 e 15x15.

4 RESULTADOS

O trabalho se iniciou com a leitura das 108 planilhas do *Microsoft Excel*, contendo os dados de todos os experimentos e criar um objeto do tipo Dataframe da biblioteca pandas. Este objeto é formado por linhas contendo os valores dos dados e colunas rotuladas, descrevendo cada tipo de coluna. Esse tipo de formato, semelhante a uma planilha, facilita a visualização dos dados e sua utilização como conjunto de entrada dos modelos desenvolvidos.

O objeto final, apresentado de forma parcial na Figura 12, possui 108 linhas e 11 colunas. Essas colunas representam os valores de bocal, pressão e quebra-jato do microaspersor, além de dados meteorológicos como a velocidade e direção do vento. Além disso, o objeto possui os valores de dia e horário no qual o experimento foi realizado e as vazões inicial, final e total observadas. Por fim, a última coluna na Figura 12 apresenta a matriz 16x16 contendo os valores de precipitação dos pluviômetros em cada um dos experimentos.

Figura 12 - Conjunto de dados pré-processado

| | bocal | pressao | quebra_jato | vel_vento | dir_vento | horario_experimento | dia_experimento | vazao_inicial | vazao_final | vazao_total | prec |
|---|-------|---------|-------------|-----------|-----------|---------------------|-----------------|---------------|-------------|-------------|---|
| 0 | 2.6 | 2.0 | 2.4 | 0.264 | 64.8 | 15.333333 | 184 | 17.9416 | 19.2245 | 1.2829 | [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.05, 0.0, ...] |
| 1 | 2.6 | 2.5 | 2.4 | 0.700 | 74.7 | 9.750000 | 156 | 17.9416 | 19.2245 | 1.2829 | [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...] |
| 2 | 2.6 | 3.0 | 2.4 | 0.312 | 130.5 | 12.800000 | 156 | 17.9416 | 19.2245 | 1.2829 | [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...] |
| 3 | 2.6 | 3.5 | 2.4 | 0.000 | 219.6 | 15.750000 | 188 | 17.9416 | 19.2245 | 1.2829 | [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.2, 0.8, 1.0, ...] |
| 4 | 2.6 | 2.0 | 2.4 | 0.280 | 89.1 | 7.666667 | 188 | 17.9416 | 19.2245 | 1.2829 | [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.05, 0.05, ...] |

Fonte: Autoria própria (2021)

Na sequência do pré-processamento dos dados foi a retirada a variável dependente contendo os valores de precipitação do conjunto de dados utilizado como entrada para os modelos desenvolvidos. Os dois objetos, contendo a variável alvo e as variáveis independentes foram separados nos conjuntos de treinamento, teste e validação.

Duas configurações para os conjuntos de treinamento e teste foram criadas, uma na qual a divisão dos dados foi realizada na proporção de 80% para treino e 20% para teste e outra onde 70% dos dados formam o conjunto de treinamento e 30% formam o conjunto de teste. Esta divisão foi realizada somente após a criação do conjunto de validação contendo, aproximadamente, 7% dos dados totais.

Os 10 objetos criados, contendo os dados de treino, teste e validação para as variáveis dependentes e independentes foram então salvos para serem utilizados em todos os experimentos realizados. A Tabela 2 apresenta os objetos desenvolvidos contendo os dados utilizados como entrada dos modelos. Além destes objetos, também foram criados mais dois arquivos contendo os dados de validação.

Tabela 2 - Conjuntos de dados utilizados como entrada dos modelos de predição

| Dados de Treinamento | Dados de Teste | Núm. de Variáveis | Núm. de Camadas |
|----------------------|----------------|-------------------|-----------------|
| 80% | 20% | 5 | 1 |
| 80% | 20% | 5 | 3 – 6 |
| 80% | 20% | 10 | 1 |
| 80% | 20% | 10 | 3 – 6 |
| 70% | 30% | 5 | 1 |
| 70% | 30% | 5 | 3 – 6 |
| 70% | 30% | 10 | 1 |
| 70% | 30% | 10 | 3 – 6 |

Fonte: Autoria própria (2021)

Essa estratégia reduz a possibilidade de que um modelo alcance resultados melhores que outros devido a uma divisão diferente e tendenciosa dos dados, já que todos os modelos utilizaram as mesmas informações em suas etapas de treinamento e teste.

4.1 VISÃO GERAL DOS RESULTADOS – MODELO UMA CAMADA

Os modelos desenvolvidos tinham como objetivo realizar a predição dos valores de água em cada um dos 256 pluviômetros utilizados na realização dos experimentos. Porém, para facilitar a visualização e apresentação dos resultados o erro foi calculado com base no valor total de água observado em cada uma das repetições e uma comparação entre o total real e o total predito pelo modelo foi realizada. A Tabela 3 apresenta a média e o desvio padrão dos valores encontrados pela aplicação das técnicas estatísticas utilizadas na avaliação do modelo com apenas uma camada obtida por meio do processo de otimização bayesiana.

Tabela 3 - Resultados de predição alcançados pelo modelo de uma camada

| Divisão dos Dados | Vars. | MAE | | MSE | | RMSE | | R2 | |
|-------------------|-------|--------------|----------|----------------|----------|--------------|----------|---------------|----------|
| | | Média | σ | Média | σ | Média | σ | Média | σ |
| 70/30 | 5 | 55,06 | 1,22 | 4619,80 | 176,45 | 67,96 | 1,30 | 0,9013 | 0,0062 |
| 80/20 | 5 | 60,06 | 2,40 | 6082,60 | 281,41 | 77,97 | 1,81 | 0,8832 | 0,0127 |
| 70/30 | 10 | 56,57 | 3,34 | 5049,35 | 463,80 | 70,99 | 3,30 | 0,8797 | 0,0364 |
| 80/20 | 10 | 50,42 | 1,84 | 4244,60 | 223,51 | 65,13 | 1,72 | 0,9234 | 0,0082 |

Fonte: Autoria própria (2021)

É possível visualizar, por meio da análise da Tabela 3, que a configuração de modelo com 10 variáveis e a divisão dos dados em 80% para treinamento e 20% para teste alcançou os melhores resultados em todas as medidas de avaliação utilizadas.

A Tabela 4 apresenta o conjunto de hiperparâmetros que alcançou os melhores resultados nas avaliações estatísticas para cada uma das configurações de divisão dos dados e do número de variáveis de entrada utilizadas na construção do modelo.

Tabela 4 - Melhores hiperparâmetros encontrados para o modelo de uma camada

| Hiperparâmetro | 70/30(5) | 80/20(5) | 70/30(10) | 80/20(10) |
|--------------------------|----------|----------|-----------|-----------|
| Algoritmo de Aprendizado | RMSprop | Adagrad | SGD | Adagrad |
| Taxa de Aprendizagem | 0,001 | 0,1 | 1,0 | 0,1 |
| Função de Ativação | Softsign | Sofsign | Sofsign | Tanh |
| Modo de Inicialização | Uniform | Uniform | Uniform | Normal |
| Taxa de Abandono | 0,0 | 0,0 | 0,2 | 0,1 |
| Taxa de Restrição | 2 | 1 | 4 | 3 |
| Número de Neurônios | 512 | 256 | 256 | 1024 |
| Função de Regularização | L1 | L2 | L2 | L1 |
| Tamanho do mini-lote | 9 | 9 | 18 | 6 |
| Número de Épocas | 500 | 600 | 500 | 500 |

Fonte: Autoria própria (2021)

A função de ativação Softsign e o modo de inicialização Uniform se destacam na análise da Tabela 4, estando presentes em três das quatro configurações analisadas. Porém, levando em consideração os resultados apresentados na Tabela 4, a configuração 80/20(10) alcançou em média os melhores resultados com utilização da função de ativação Tanh e o modo de inicialização Normal.

4.2 VISÃO GERAL DOS RESULTADOS – MODELO VÁRIAS CAMADAS

Utilizando os conjuntos de hiperparâmetros que alcançaram os melhores resultados na análise estatística, apresentados na Tabela 4, uma nova avaliação foi realizada para verificar se existe uma melhora na eficiência das predições com a utilização de modelos com várias camadas. A Tabela 5 apresenta os resultados desse processo.

Tabela 5 - Resultados de predição alcançados pelos modelos de várias camadas

| Divisão dos Dados | Camadas | Variáveis | MAE | MSE | RMSE | R2 |
|-------------------|---------|-----------|--------------|----------------|--------------|---------------|
| 70/30 | 2 | 5 | 54,32 | 4411,76 | 66,42 | 0,9074 |
| 80/20 | 2 | 5 | 69,74 | 7518,23 | 86,70 | 0,8623 |
| 70/30 | 3 | 10 | 51,66 | 4226,40 | 65,01 | 0,9103 |
| 80/20 | 2 | 10 | 57,36 | 7269,71 | 85,26 | 0,8569 |

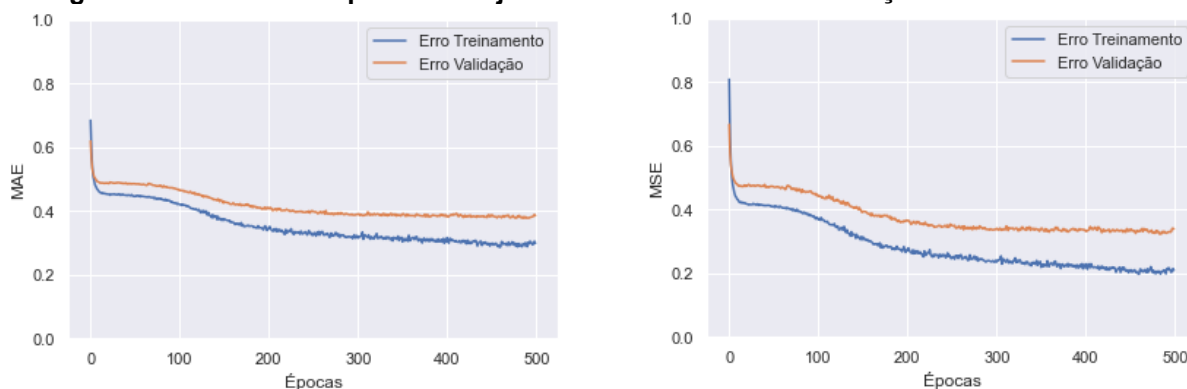
Fonte: Autoria própria (2021)

É possível verificar, por meio da análise da Tabela 5, que os melhores resultados encontrados nos experimentos de múltiplas camadas utilizando a configuração de divisão dos dados na proporção de 70/30 e com a utilização de 10 variáveis. Utilizando três camadas ocultas, esse modelo alcançou um RMSE de 65,01 e um R^2 de 0,9103.

4.3 ANÁLISE DO RESULTADOS – MELHOR MODELO

Um modelo utilizando a divisão dos dados na proporção 80/20 e 10 variáveis de entrada registrou, de modo geral, os melhores resultados quando comparado às outras configurações analisadas. Esse modelo alcançou um RMSE de 64,2487 e um R^2 de 0,9287. O conjunto de hiperparâmetros desse modelo foi apresentado na Tabela 4. Esse modelo também possui uma boa capacidade de generalização, o que pode ser confirmado por meio da análise de comparação dos erros, apresentada na Figura 13.

Figura 13 – MAE e MSE para os conjuntos de treinamento e validação do melhor modelo

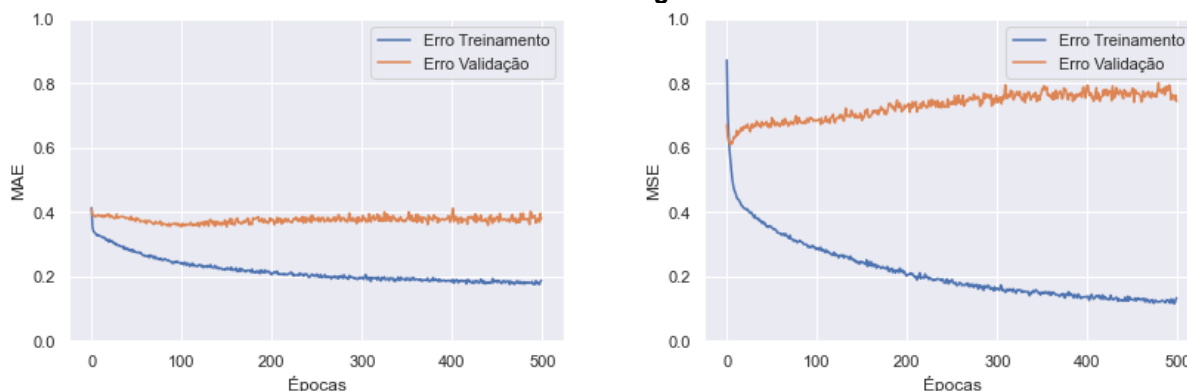


Fonte: Autoria própria (2021)

É possível observar na Figura 13 que não aparenta estar havendo *overfitting* no modelo analisado já que o erro de validação está próximo do erro de treinamento durante todo o processo de treinamento.

Outros modelos avaliados, porém, alcançaram resultados melhores quando comparados ao modelo analisado nesta seção. Por exemplo, um modelo utilizando a proporção de divisão dos dados 80/20 e 10 variáveis de entrada alcançou um RMSE de 62,55 e um R^2 de 0,9620. Porém, uma análise dos erros de validação e treinamento desse modelo, apresentada na Figura 14, demonstra que este não é a melhor escolha para o problema proposto.

Figura 14 – MAE e MSE dos conjuntos de treinamento e validação de um modelo com *overfitting*



Fonte: Autoria própria (2021)

É possível visualizar na Figura 14 que a curva de generalização do modelo analisado possui um padrão de *overfitting*, tendo se adaptado muito bem aos dados do conjunto de treinamento e possuindo dificuldades em prever novos dados nunca antes vistos.

É possível verificar, tanto nos modelos com apenas uma camada quanto nos modelos de várias camadas, que quanto maior o valor de R^2 alcançado menor são os erros associados, indicando que modelos com alta generalização produzem erros de predição menores.

Utilizando os valores de precipitação dos experimentos do conjunto de validação e os valores de precipitação preditos, o cálculo do coeficiente de uniformidade de Christiansen (CUC) em três distâncias, em metros, diferentes foi realizado para verificar se o modelo selecionado é capaz de realizar boas predições dessa variável. A Tabela 6 apresenta os resultados desse processo.

Tabela 6 - Comparação entre os valores reais e os valores preditos de CUC

| Experimento | Distância 12x12 | | | Distância 12x15 | | | Distância 15x15 | | |
|-------------|-----------------|------------|------|-----------------|------------|-------|-----------------|------------|-------|
| | Valor Predito | Valor Real | Erro | Valor Predito | Valor Real | Erro | Valor Predito | Valor Real | Erro |
| 23 | 94,21 | 94,04 | 0,17 | 96,02 | 96,21 | -0,18 | 94,73 | 95,24 | -0,51 |
| 36 | 88,77 | 86,91 | 1,85 | 91,01 | 88,61 | 2,40 | 93,50 | 91,5 | 1,99 |
| 48 | 87,56 | 87,03 | 0,52 | 89,16 | 88,51 | 0,64 | 91,92 | 90,99 | 0,93 |
| 56 | 87,89 | 87,42 | 0,46 | 89,76 | 89,51 | 0,25 | 92,05 | 91,86 | 0,18 |
| 76 | 88,57 | 87,35 | 1,22 | 91,11 | 89,31 | 1,79 | 93,77 | 92,17 | 1,59 |
| 85 | 87,56 | 86,68 | 0,88 | 89,73 | 89,27 | 0,46 | 92,03 | 90,94 | 1,08 |
| 91 | 88,78 | 88,12 | 0,65 | 91,26 | 90,48 | 0,78 | 94,11 | 93,05 | 1,05 |
| 104 | 88,75 | 88,11 | 0,64 | 91,63 | 90,4 | 1,19 | 94,26 | 93,01 | 1,24 |

Fonte: Autoria própria (2021)

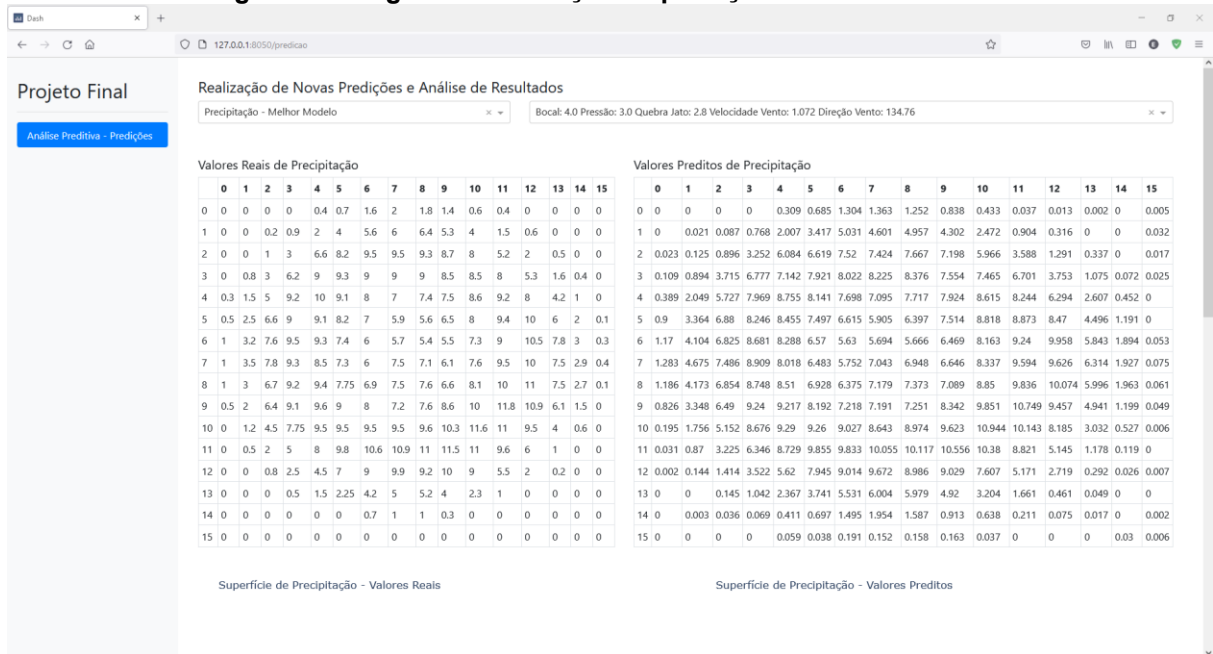
A Tabela 6 apresenta os valores de CUC observados para cada um dos experimentos que foram o conjunto de dados de validação para as distâncias, em metros, de 12x12, 12x15 e 15x15. Este conjunto representa por volta de 7% do conjunto de dados original e que foi separado para realizar validações após as fases de treinamento e teste dos modelos desenvolvidos.

Também são apresentados nessa tabela os valores de CUC para as predições de precipitação realizadas pelo melhor modelo selecionado para as três distâncias. Os melhores resultados alcançados foram na predição do CUC na distância 12x12, com o modelo selecionado tendo um erro médio de 0,79 e um desvio padrão de 0,52.

O ambiente *web*, desenvolvido com o auxílio da plataforma Plotly Dash, foi utilizado para facilitar a visualização por meio da comparação dos resultados alcançados pelo melhor modelo de predição e os valores reais dos experimentos

que compõe o conjunto de validação. A Figura 15 apresenta a página de realização das predições desenvolvida neste ambiente.

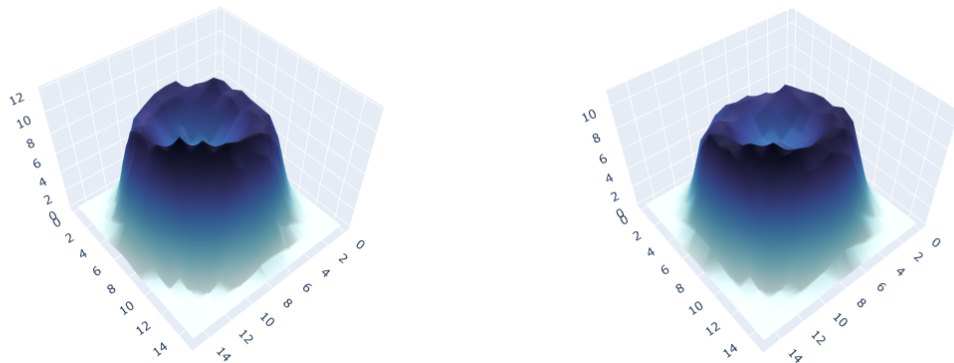
Figura 15 - Página de realização de predições do ambiente web



Fonte: Autoria própria (2021)

Além da comparação dos valores de precipitação esta página também foi utilizada para simular a superfície de precipitação dos valores reais observados nos experimentos e compará-los com os valores de predição alcançados pelo modelo. A Figura 16 apresenta essa comparação.

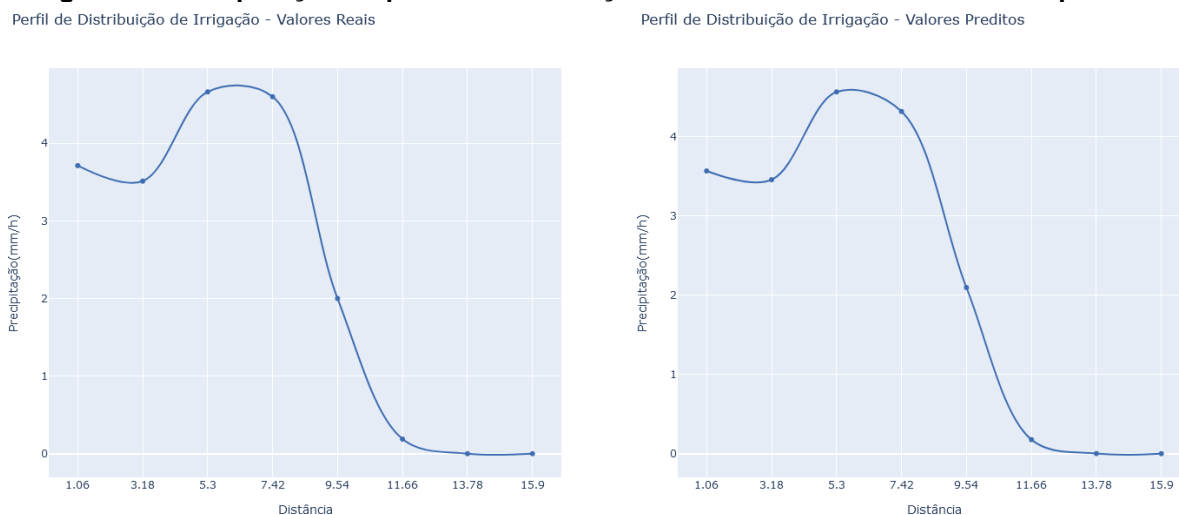
Figura 16 - Comparação da superfície de precipitação dos valores reais e dos valores preditos
Superfície de Precipitação - Valores Reais Superfície de Precipitação - Valores Preditos



Fonte: Autoria própria (2021)

Nesse mesmo ambiente também foi feita a comparação do perfil de distribuição da irrigação utilizando os valores reais do experimento e o perfil de distribuição observado com a utilização dos valores de precipitação preditos pelo modelo. Essa comparação é apresentada na Figura 17.

Figura 17 - Comparação do perfil de distribuição dos valores reais e dos valores preditos



Fonte: Autoria própria (2021)

Tanto a comparação da superfície de precipitação, apresentada na Figura 16, quanto a comparação dos perfis de distribuição, apresentada na Figura 17, utilizam os valores de precipitação observados no experimento 91. Ambas as comparações demonstram a capacidade do modelo em simular corretamente essas propriedades tão importantes para a análise de sistemas de irrigação.

4.4 DISCUSSÃO DOS RESULTADOS

A arquitetura de modelo de predição com apenas uma camada foi responsável por alcançar os melhores resultados em relação aos modelos analisados. Utilizando a arquitetura de divisão dos dados na proporção 80/20 com 10 variáveis de entrada, os modelos de uma camada alcançaram um RMSE média de 65,13 e um R^2 0,9234. Por meio destes valores é possível verificar a boa capacidade do modelo em realizar predições da precipitação de microaspersores.

O melhor modelo geral analisado também utilizou a divisão dos dados de 80/20 com 10 variáveis. O conjunto de hiperparâmetros, apresentado na Tabela 4, foi utilizado para construir um modelo que alcançou um RMSE de 64,2487 e um R^2

de 0,9287. Apesar de outros modelos desenvolvidos terem alcançado valores melhores na etapa de análise estatística do projeto, este foi o modelo com os melhores resultados e que não aparentou estar havendo *overfitting* durante seu processo de treinamento. Isso garante a capacidade de generalização do modelo para previsões futuras.

Porém, como a utilização de mais variáveis pode acarretar em um aumento na dificuldade de execução dos experimentos e, como os modelos com cinco variáveis de entrada alcançaram valores próximos de R^2 ao melhor modelo analisado, a utilização de um modelo com menos variáveis onde a coleta de dados é mais fácil pode ser considerada uma escolha válida.

Em sua pesquisa para simular a precipitação de microaspersores com a ajuda de redes neurais artificiais, Menezes et. al. (2015) foi capaz de alcançar uma forte correlação entre os valores observados e os valores simulados. Utilizando dados do conjunto de validação, todos os experimentos obtiveram um valor acima de 0,95 nos testes de correlação de Spearman.

Já em Wolfrann (2018) modelos matemáticos foram utilizados em conjunto com redes neurais artificiais na estimativa da precipitação na irrigação por aspersão. Este projeto também alcançou bons resultados, com um coeficiente de determinação (R^2) em média igual a 0,93 para o ajuste linear.

Utilizando os hiperparâmetros definidos nos melhores modelos de uma camada para cada uma das configurações de divisão dos dados e número de variáveis de entrada foram construídos novos modelos onde o número de camadas foi testado como um hiperparâmetro. Este processo foi realizado para verificar se os resultados alcançados podem ser melhorados com uma arquitetura profunda.

Por meio da comparação entre a Tabela 3 e a Tabela 5 é possível verificar que apenas a configuração com a divisão dos dados na proporção de 70/30 utilizando cinco variáveis de entrada alcançou resultados superiores, possuindo um valor de R^2 levemente maior quando comparado com o mesmo modelo de apenas uma camada. Os valores de R^2 em todas as outras configurações analisadas foram inferiores em comparação às mesmas arquiteturas no modelo raso. A utilização de mais camadas ocultas na construção do modelo de previsão desenvolvido neste projeto não trouxe melhoras para os resultados alcançados.

Porém, a utilização de técnicas de *Deep Learning* se mostrou bem-sucedida quando aplicadas a solução do problema proposto. O melhor modelo observado

utilizou o algoritmo de otimização Adagrad, uma solução mais moderna se comparada aos métodos clássicos como o Gradiente Descendente Estocástico. Isso demonstra que a utilização de novas técnicas de aprendizado de máquina, em conjunto com as ferramentas já bem definidas na literatura, pode trazer resultados concretos para a resolução de problemas de áreas onde a aplicação desse tipo de ferramenta ainda é limitada.

O melhor modelo de predição analisado possui boa capacidade de generalização, demonstrada pelo seu valor R^2 de 0,9287 alcançado. Além disso, este modelo foi capaz de simular corretamente a superfície de irrigação, como pode ser constatado pela observação da Figura 16, e produzir um perfil de distribuição próximo do perfil real dos aspersores analisados, como pode ser visto na Figura 17. Conclui-se, portanto, que a metodologia apresentada neste trabalho pode servir de apoio à tomada de decisão por parte dos gerenciadores devido a sua capacidade de simulação de variáveis e processos importantes para o desenvolvimento e manutenção de sistemas de sistemas de irrigação por microaspersão.

5 CONCLUSÃO

Esta pesquisa se propôs a realizar o desenvolvimento de modelos de predição baseados em redes neurais que sejam capazes de estimar o coeficiente de uniformidade de microaspersores. Estes modelos foram construídos utilizando a biblioteca Keras e as técnicas de otimização de hiperparâmetros com busca bayesiana e validação cruzada.

A arquitetura do melhor modelo encontrado com a execução deste processo possui apenas uma camada oculta e uma divisão de 80% dos dados, formados por 108 experimentos utilizando dados de precipitação e dados meteorológicos, para o conjunto de treinamento e 20% para o conjunto de teste. Além disso foram utilizadas 10 variáveis de entrada. Esta configuração alcançou um RMSE de 64,2487 e um R^2 de 0,9287. Outros testes foram realizados com arquiteturas profundas, mas estes experimentos não demonstraram a melhora nos resultados alcançados com a utilização de mais de uma camada oculta.

O melhor modelo desenvolvido foi capaz de realizar boas predições do coeficiente de uniformidade de Christiansen (CUC). Utilizando o conjunto de dados

de validação, composto 7% dos dados originais e que foram escolhidos aleatoriamente antes da divisão dos conjuntos de treinamento e teste, o modelo teve um erro médio de 0,79 nos valores de CUC utilizando a distância de 12x12m.

Utilizando a biblioteca Plotly Dash, uma aplicação *web* foi desenvolvida para apresentar informações importantes no gerenciamento de sistemas de irrigação como gráficos de simulação da superfície de irrigação e do perfil de distribuição da água. Realizando mais uma vez a comparação entre os valores preditos pelo melhor modelo e os valores reais do conjunto de validação, foi possível verificar que boas previsões foram realizadas e que a solução proposta nesta pesquisa pode ser utilizada por tomadores de decisão na construção e otimização de novos sistemas de irrigação, ajudando a diminuir os custos relacionados à execução de experimentos caros para estimar variáveis como o coeficiente de uniformidade.

5.1 TRABALHOS FUTUROS

Como possibilidade de novas pesquisas que podem contribuir no enriquecimento do trabalho desenvolvido podem ser citadas:

1. Utilização de um conjunto de dados maior para verificar se existe uma mudança nos resultados dos modelos com múltiplas camadas.
2. Verificar a capacidade de predição do modelo desenvolvido em outros coeficientes de uniformidade, como o coeficiente de uniformidade de distribuição (CUD) e o coeficiente de uniformidade estatístico (CUE).
3. Melhoria do ambiente *web* desenvolvido, onde os usuários possam enviar novos dados de experimentos realizados para serem analisados.

REFERÊNCIAS

- ADENUGBA, F. et al. Smart irrigation system for environmental sustainability in Africa: An Internet of Everything (IoE) approach. **Mathematical Biosciences and Engineering**, Ota, v. 16, n. 5, p. 5490-5503, Jun 2019. ISSN 1547-1063.
- Alves, C. F. G.; Lucio, J. C. B.; Menezes, S. M. de; Almeida, R. S. de; Silva, F. B. da; Silva, J. C. da. **Avaliação de um sistema de irrigação por microaspersão na cultura do maracujá (Passiflora sp.)**. XXV CONIRD - Congresso Nacional de Irrigação e Drenagem. São Cristóvão 2015. p. 1221-1226.
- ANTUNES, J. A. **Coletânea particular de fotos de irrigação**. Caçador. 2006.
- ASAWA, G. L. **Irrigation and water resources engineering**. Nova Deli: New Agen International Limited, 2008.
- BARBEDO, J. G. A. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. **Computers and Electronics in Agriculture**, Campinas, v. 153, p. 46-53, Ago 2018. ISSN 0168-1699.
- BARGOS, F. F.; LAMAS, W. D. Q.; BILATO, G. A. Computational tools and operational research for optimal design of co-generation systems. **Renewable and Sustainable Energy Reviews**, Lorena, v. 93, p. 507-516, Outubro 2018. ISSN 1364-0321.
- BELLIDO, B. S.; MARTÍNEZ, J. M.; URRESTARAZU, L. P. Development of an automatic test bench to assess sprinkler irrigation uniformity in different wind conditions. **Computer and Electronics in Agriculture**, Chipiona, v. 151, p. 31-40, Mai 2019. ISSN 168-1699.
- BELYADI, H.; HAGHIGHAT, A. **Machine learning guide for oil and gas using Python**. Cambridge: Gulf Professional Publishing, 2021.
- BERNARDO, S.; SOARES, A. A.; MANTOVANI, E. **Manual de Irrigação**. 8. ed. Viçosa: Ufv, 2009.
- BRALTS, V. F.; PANDEY, S. R.; MILLER, A. Energy savings and irrigation performance of a modified center pivot irrigation system. **Applied Engineering in Agriculture**, v. 10, n. 1, p. 27-36, 1994. ISSN 0883-8542.
- BROWNLEE, J. **Deep learning with Python - Develop deep learning models on Theano and TensorFlow using Keras**. 1.8. ed. , 2017.
- CAI, Y.; ZHENG, W.; ZHANG, X.; ZHANGZHONG, L.; XUE, X Research on soil moisture prediction model based on deep learning. **PloS one**, v. 14, n. 4, p. 1-19, Abril 2019. ISSN 1932-6203.
- CHLINGARYAN, A.; SUKKARIEH, S.; WHELAN, B. Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. **Computers and Electronics in Agriculture**, v. 151, p. 61-69, Mai 2018. ISSN 0168-

1699.

CHOLLET, F. About Keras. **Keras**, 2015. Disponível em: <<https://keras.io/about/>>. Acesso em: 20 Junho 2021.

CHRISTIANSEN, J. E. **Irrigation by sprinkling**. Berkeley: Berkeley, Cal.: Agricultural Experiment Station, v. B670, 1942.

DANNER, M.; BERGER, K.; WOCHER, M.; MAUSER, W.; HANK, T. Efficient RTM-based training of machine learning regression algorithms to quantify biophysical & biochemical traits of agricultural crops. **ISPRS Journal of Photogrammetry and Remote Sensing**, Munich, v. 173, p. 278-296, Jan 2021. ISSN 0924-2716.

EL BILABI, A.; TALEB, A.; BROUZUIYNE. Groundwater quality forecasting using machine learning algorithms for irrigation purposes. **Agricultural Water Management**, v. 245, p. 1-13, Fev 2021. ISSN 0378-3774.

FAO. **How to feed the world 2050**. High-Level Expert Forum. Rome: Office of the Director, Agricultural Development Economics Division. 2009. p. 4.

FAO. **The future of food and agriculture - Trends and challenges**. Rome, 2017.

FARIA, L. C. **Modelagem da aplicação de água de aspersores de tamanho médio em condições de vento**. Universidade Federal de Lavras. Lavras, p. 91. 2011.

FAUSETT, L. **Fundamental of neural networks: architectures, algorithms and applications**. 1994.

FERREIRA, A. D. S.; FREITAS, D. M.; DA SILVA, G. G.; PISTORI, H.; FOLHER, M. T. Unsupervised deep learning and semi-automatic data labeling in weed discrimination. **Computer and Electronics in Agriculture**, v. 165, p. 1-11, Ago 2019. ISSN 0168-1699.

FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS. **Declaration of the world summit on food security**. Food and Agriculture Organization of the United Nations. Rome, p. 1-7. 2009.

GARCÍA-SALDANA, A; LANDEROS-SÁNCHEZ, C; CASTAÑEDA-CHÁVEZ, M. R.; MARTÍNEZ-DÁVILA, J. P; PÉREZ-VÁZQUEZ, A; CARRILLO-ÁVILA, E. Fertirrigation with Low-Pressure Multi-Gate Irrigation Systems in Sugarcane Agroecosystems: A Review. **Pedosphere**, Tepetates, v. 29, n. 1, p. 1-11, Fev 2019. ISSN 1002-0160.

GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de pesquisa**. 1 ed. ed. Porto Alegre: Editora da UFRGS, 2009.

GÉRON, A. **Hands-on machine learning with scikit-learn, Keras, and TensorFlow**. 2ª. ed. Sebastopol: O'Reilly Media, Inc., 2019.

GONZÁLEZ, R. P.; CAMACHO, E. P.; MONTESINOS, P.; RODRÍGUEZ, D. J. A.

Prediction of applied irrigation depths at farm level using artificial intelligence techniques. **Agricultural Water Management**, Córdoba, v. 206, p. 229-240, Jul 2019. ISSN 0378-3774.

GOODFELLOW, I.; BENGUI, Y.; COURVILLE, A. **Deep learning**. MIT Press, 2016.

HADDADI, A; HOSSEINI, A.; JOHANSEN, A; OLSSON, N. Pursuing Value Creation in Construction by Research - A Study of Applied Research Methodologies. **Procedia Computer Science**, Trondheim, v. 121, p. 1080-1087, Novembro 2017. ISSN 1877-0509.

HAYKIN, S. **Neural networks and learning machines**. 3^a. ed. Hamilton: Person Prentice Hall, 1999.

KARANDISH, F.; SIMUNEK, J. A comparison of numerical and machine-learning modeling of soil water content with limited input data. **Journal of Hydrology**, Zabol, v. 543, p. 892-909, Dez 2016. ISSN 0022-1694.

KELLER, J.; BLIESNER, R. D. **Sprinkle and trickle irrigation**. Springer US, 1990.

KUMAR, D.; ADAMOWSKI, J.; SURESH, R.; OZGA-ZIELINSKI, B. Estimating Evapotranspiration Using an Extreme Learning Machine Model: Case Study in North Bihar, India. **Journal of Irrigation and Drainage Engineering**, Roorkee, v. 142, n. 9, p. 1-9, Set 2016. ISSN 0733-9437.

LUGER, G. F.; STUBBLEFIELD, W. A. **Artificial Intelligence: structures and strategies for complex problem solving**. 2. ed. Benjamin/Cummings, 1993.

MAROUFPOOR, S.; SHIRI, J.; MAROUFPOOR, E. Modeling the sprinkler water distribution uniformity by data-driven methods based on effective variables. **Agricultural Water Management**, Tehran, v. 215, p. 63-73, Jan 2019. ISSN 0378-3774.

MENEZES, P. L.; AZEVEDO, C. A. V.; EYING, E.; NETO, J. D.; LIMA, V. L. A. Artificial neural network model for simulation of water distribution in sprinkle irrigation. **Revista Brasileira de Engenharia Agrícola e Ambiental**, Campina Grande, v. 19, p. 817-822, Set 2015. ISSN 1807-1929.

MERONI, M. et al. Yield forecasting with machine learning and small data: What gains for grains? **Agricultural and Forest Meteorology**, Ispra, v. 1-12, n. 308-309, p. 308-30, Jul 2021. ISSN 0168-1923.

MITCHELL, T. M. **Machine learning**. McGraw-Hill, 1997.

MOHAPATRA, A. G.; LENKA, S. K.; KESWANI, B. Neural Network and Fuzzy Logic Based Smart DSS Model for Irrigation Notification and Control in Precision Agriculture. **Proceedings of the National Academy of Sciences, India Section A: Physical Sciences**, v. 89, n. 1, p. 67-76, Jan 2019. ISSN 0369-8203.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of machine**

learning. Cambridge: The MIT Press, 2012.

MONTERO, J.; TARJUELO, J.; CARRIÓN, P. SIRIAS: a simulation model for sprinkler irrigation: I. Description of model. **Irrigation Science**, Albacete, v. 20, n. 2, p. 73-84, Jun 2001. ISSN 0342-7188.

NAWANDAR, N. K.; SATPUTE, V. R. IoT based low cost and intelligent module for smart irrigation system. **Computer and Electronics in Agriculture**, Nagpur, v. 162, p. 979-990, Mai 2019. ISSN 0168-1699.

PANDEY, S. The Role of Irrigation for Food Security and Sustainability. In: FERRANTI, P.; BERRY, M. E.; ANDERSON, R. J. **Encyclopedia of Food Security and Sustainability**. Manila: Elsevier, v. 3, 2019. Cap. 20, p. 143-146.

PATTERSON, J.; GIBSON, A. **Deep learning: a practitioner's approach**. Sebastopol: O'Reilly Media, 2017.

PEREA, R. G.; POYATO, E. C.; MONTESINOS, P.; DÍAS, J. A. R. Optimisation of water demand forecasting by artificial intelligence with short data sets. **Biosystems Engineering**, Córdoba, v. 177, p. 59-66, Jan 2018. ISSN 1537-5110.

PHOCAIDES, A. **Handbook of pressurized irrigation techniques**. 2. ed. Rome, 2007.

Qi, J.; Du, J.; Siniscalchi, S. M.; Ma, X.; Lee, C. On Mean Absolute Error for Deep Neural Network Based Vector-to-Vector Regression. **IEEE Signal Processing Letters**, 12 Ago 2020. 1-5.

RAZAVIAN, M.; PAECH, B.; TANG, A. Empirical research for software architecture decision making: An analysis. **Journal of Systems and Software**. Eindhoven, v. 149, p. 360-381, Março 2019. ISSN 0164-1212.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. Englewood: Prentice-Hall, 1995.

SAYARI, S.; MAHDAVI-MEYMAND, A.; ZOUNEMAT-KERMANI, M. Irrigation water infiltration modeling using machine learning. **Computers and Electronics in Agriculture**, v. 188, p. 1-9, Jan 2021. ISSN 0168-1699.

SCHALKOFF, R. I. **Artificial intelligence: an engineering approach**. Nova York: McGraw-Hill, 1990.

SEGINER, I.; NIR, D.; BERNUTH, R. D. V. Simulation of Wind-Distorted Sprinkler Patterns. **Journal of Irrigation and Drainage Engineering**, v. 117, n. 2, Mar 1991. ISSN 0733-9437.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding machine learning: from theory to algorithms**. Nova York: Cambridge University Press, 2014.

SHEIKHESMAEILI, O.; MONTERO, J.; LASERNA, S. Analysis of water application

with semi-portable big size sprinkler irrigation systems in semi-arid areas.

Agricultural Water Management, Ahvaz, v. 163, p. 275-284, Jan 2016. ISSN 0378-3774.

SINGH, D.; SINGH, B. Investigating the impact of data normalization on classification performance. **Applied Soft Computing Journal**, Longowal, v. 97, p. 1-23, Dez 2020. ISSN 1568-4946.

SORRENTINO, M. B.; PÉREZ, R. **New Jersey irrigation guide**. Somerset, 2005.

STEURER, M.; HILL, R. J.; PFEIFER, N. Metrics for evaluating the performance of machine learning based automated valuation models. **Journal of Property Research**, Graz, v. 38, n. 2, p. 99-129, Nov 2020. ISSN 1466-4453.

TESTEZLAF, R. **Irrigação: métodos, sistemas e aplicações**. Campinas: [s.n.], 2017.

USDA. **Irrigation guide**. 1997.

VANDERPLAS, J. **Python data science handbook - Essential tools for working with data**. 1^a. ed. Sebastopol: O'Reilly Media, v. 1, 2017.

VOLSCHENK, T. Water use and irrigation management of pomegranate trees - A review. **Agricultural Water Management**, Stellenbosch, v. 241, p. 1-14, Nov 2020. ISSN 0378-3774.

WOLFRANN, J. **Redes neurais artificiais para estimar a precipitação na irrigação por aspersão**. Dissertação (Mestrado em Tecnologias Computacionais para o Agronegócio) Universidade Tecnológica Federal do Paraná. Medianeira, p. 100. 2018.

WILLMOTT, C. J.; MATSUURA, K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. **Climate Research**, Newark, v. 30, p. 79-82, Dez 2005. ISSN 0936-577X.

WU, J.; CHEN, X.; ZHANG, H.; XINONG, L.; HANG, L.; SI-HAO, D. Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. **Journal of Electronic Science and Technology**, Chengdu, v. 17, n. 1, p. 26-40, Mar. 2019. ISSN 1674-862X.

YANG, L.; SHAMI, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. **Neurocomputing**, Ontario, v. 415, p. 295-316, Nov. 2020. ISSN 0925-2312.

ZHANG, A. et al. **Dive into deep learning**. , 2021.