

Estimativa do Coeficiente de Uniformidade de Microaspersores por Meio da Aplicação de Técnicas de Redes Neurais Artificiais

Everton Schneider dos Santos
PPGTCA - UTFPR
Medianeira, Paraná. Brasil
everton_schneider1@outlook.com

Arnaldo Candido Junior
PPGTCA - UTFPR
Medianeira, Paraná. Brasil
arnaldocan@gmail.com

Paulo Lopes de Menezes
PPGTCA - UTFPR
Medianeira, Paraná. Brasil
plopes@utfpr.edu.br

Abstract—The objective of this study was to evaluate the capacity of artificial neural networks to estimate distribution uniformity coefficient of micro sprinklers. The following features of a micro sprinkler model Pingo 360° brand Fabrimar were observed: pressure (kgf/cm³), nozzle (mm), jet breakup, wind average speed (m/s) and direction (degrees), initial flow rate, final flow rate, total flow rate, time and date of the experiment. In total, 256 rain gauges scattered around the micro sprinkler were used to measure how much water were spent during irrigation. Using Bayesian search and hyper parameter optimization techniques, an artificial neural network model, capable of estimate the Christiansen Uniformity Coefficient, was created. Using a distance of 12x12 meters between sprinklers, this model achieved an R² of 92.87%, demonstrating that the methodology applied in this work is capable of simulate the irrigation process of the micro sprinkler used during the experiments.

Resumo—O objetivo deste trabalho foi avaliar a capacidade de redes neurais artificiais em estimar o coeficiente de uniformidade da irrigação realizada por microaspersores. As seguintes características do aspersor Pingo giro completo 360° da marca Fabrimar foram observadas: pressão (kgf/cm³), bocal (mm), quebra jato, velocidade média (m/s) e direção do vento (graus), vazão inicial, vazão final, vazão total, horário e data do experimento. Uma malha de 256 pluviômetros, dispersa ao redor do microaspersor, foi utilizada para medir os valores de água gastos durante a irrigação. Utilizando técnicas de busca Bayesiana e otimização de hiperparâmetros, um modelo de rede neural artificial foi desenvolvido com a capacidade de estimar o Coeficiente de Uniformidade de Christiansen. Em uma distância entre aspersores de 12x12 metros, este modelo alcançou um R² de 92,87%, demonstrando que a metodologia aplicada neste trabalho é capaz de simular o processo de irrigação do microaspersor utilizado nos experimentos.

Palavras-chave—agricultura irrigada; perfil de distribuição; inteligência artificial.

I. INTRODUÇÃO

Os processos de irrigação contribuem diretamente para a segurança alimentar. Seja pelo aumento da produtividade ou

pela expansão da área de cultivo, quando a insuficiência de chuva limita a área plantada, diversos aspectos da agricultura são afetados favoravelmente pela irrigação [1].

Sistemas de irrigação por microaspersão utilizam pequenos emissores e possuem uma capacidade de irrigação maior quando comparado a outros sistemas devido a sua vazão superior [2]. Entre as vantagens desse sistema podem ser citadas a sua facilidade de instalação, flexibilidade e facilidade de gerenciamento [3].

Apesar disso, sistemas de irrigação por microaspersão são altamente influenciados por variáveis climáticas. Tais fatores podem causar perda parcial de água por evaporação e desvio para fora da área de irrigação [4].

Métodos como 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 [5].

Na fórmula de medição do coeficiente de Christiansen, apresentada na Equação 1, o x_i representa a profundidade da água em cada coletor utilizado, \bar{x} representa 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

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

Seguindo a metodologia definida em Alves et al. (2015), a Tabela I apresenta a classificação de desempenho para o CUC.

A capacidade de modelar a uniformidade de distribuição possui um papel importante na utilização otimizada da água na irrigação [6]. Modelos de simulação computacional, como as redes neurais artificiais, podem ser utilizados na irrigação

TABELA I
CRITÉRIOS DE DESEMPENHO DO CUC

Classificação	CUC (%)
Excelente	>90
Bom	80 – 90
Razoável	70 – 80
Ruim	60 – 70
Inaceitável	<60

com o objetivo de evitar testes de campo trabalhosos e para projetar e melhorar sistemas de irrigação.

Diversos estudos da aplicação de técnicas de inteligência artificial na simulação de processos da irrigação podem ser encontrados na literatura. [7] obteve bons resultados utilizando técnicas de aprendizado de máquina para estimar a infiltração de água em sistemas de irrigação por sulcos. Neste trabalho, modelos como redes neurais artificiais, regressão linear multivariada e seus respectivos modelos integrativos combinados com o algoritmo de vagalume foram utilizados para estimar a água infiltrada. Os modelos de redes neurais artificiais forneceram ajustes adequados aos dados observados, com resultados para o coeficiente de determinação (R^2) acima de 0,95.

Em seu trabalho, [8] utilizou uma abordagem de aprendizado de máquina para estimar, no curto prazo, a umidade do solo para o cultivo de batata. Modelos baseados em técnicas como redes neurais artificiais, florestas aleatórias e máquinas de vetor de suporte foram criadas para realizar a predição do potencial hídrico de uma plantação de batatas em até uma semana. Todos os modelos forneceram ajustes adequados aos dados observados, com valores de R^2 acima de 0,8.

Este trabalho tem como objetivo 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, minimizando a utilização dos recursos hídricos sem interferir na produção agrícola. Este conhecimento pode ajudar gestores na projeção de sistemas de irrigação onde a utilização de recursos hídricos seja minimizada sem interferir na produção agrícola.

II. MATERIAL E MÉTODOS

Os dados utilizados nesta pesquisa foram coletados por [9]. Os experimentos foram executados na fazenda Ouro Verde, localizada no município de Medianeira, Brasil (25°12'19,26"S; 54°3'27,93"W; 360m). Uma malha contendo 256 pluviômetros com um microaspersor no centro foi utilizada para medir a quantidade de água gasta durante a irrigação. Os pluviômetros foram posicionados a uma distância de 1,5m um do outro e a uma altura de 1,25m em relação à superfície.

Características de diâmetro do bocal (mm) e quebra jato do microaspersor Pingo giro completo 360° foram observadas durante os experimentos realizados. Os valores testados de diâmetro de bocal e quebra jato foram, respectivamente: 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/cm^3) diferentes: 2.0, 2.5, 3.0 e 3.5.

Outros dados coletados incluem a velocidade média (m/s) e direção do vento (graus), além de valores de vazão do microaspersor e data e horário da execução dos experimentos. Foram realizadas três repetições para cada um dos conjuntos de características do microaspersor, totalizando 27 ensaios com duração de duas horas cada.

Por volta de 7% dos dados pré-processados foram separados como conjunto de teste dos modelos. O restante dos dados foram divididos nas proporções 70/30 e 80/20 entre treinamento e validação, respectivamente. Foram definidas duas configurações de variáveis de entrada a serem utilizadas pelos modelos de predição construídos. A primeira configuração utiliza as 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, as variáveis horário do experimento, número de dias transcorridos em relação a 1° de Janeiro do ano do experimento, vazão inicial, final e total do microaspersor.

Dois arquiteturas de modelos de predição foram desenvolvidas utilizando o Keras¹, uma API de Deep Learning escrita na linguagem de programação Python. A primeira arquitetura possui uma camada oculta e a segunda arquitetura possui de três a seis camadas ocultas. Ambas as arquiteturas possuem uma camada de entrada com cinco ou dez neurônios, dependendo do número de variáveis avaliadas e uma camada de saída com 256 neurônios, representando o número de pluviômetros utilizados na coleta de dados.

A construção de uma rede neural artificial é um processo complexo que envolve a determinação de um algoritmo de aprendizagem e a obtenção de uma arquitetura otimizada por meio do ajuste de um conjunto de hiperparâmetros. Esse tipo de parâmetro não pode ser estimado durante a aprendizagem e deve ser definido antes do treinamento do modelo [10].

A busca Bayesiana é um tipo de técnica de otimização de hiperparâmetros que utiliza um modelo preditivo para modelar o espaço de busca e alcançar uma boa combinação de valores dos hiperparâmetros o mais rápido possível, já que uma busca exaustiva em todas as combinações se tornaria muito complexa em um espaço de pesquisa com inúmeros hiperparâmetros [11].

¹<https://keras.io/>

Nesta pesquisa, uma busca Bayesiana foi executada com o objetivo de encontrar um conjunto de hiperparâmetros otimizado para o modelo de predição. Os hiperparâmetros avaliados incluem diferentes otimizadores [12], algoritmos de transformação dos dados [13], funções de ativação [14] e métodos para inicialização dos pesos [15]. Os valores para os hiperparâmetros são apresentados na Tabela II.

TABELA II
HIPERPARÂMETROS GERAIS AVALIADOS

Hiperparâmetros	Valores
Otimizador	Adam, SGD, RMSprop, Adagrad
Mini-lote	6, 9, 18
Épocas	400, 500, 600, 700
Neurônios	128, 256, 512, 768, 1024
Transformação	Standard, MinMax, Robust, Power, Quantile Uniform, Quantile Normal

Outros hiperparâmetros utilizados pela busca Bayesiana incluem: taxa de aprendizado, taxa de *dropout*, taxa de restrição dos pesos e função de regularização. A Tabela III apresenta os valores avaliados para estes hiperparâmetros.

TABELA III
HIPERPARÂMETROS DE TREINAMENTO AVALIADOS

Hiperparâmetros	Valores
Função de ativação	GeLu, ReLu, Swish, Mish, Sigmoid, Softplus, Softsign, Tanh
Taxa de aprendizado	1.0, 0.1, 0.001, 0.001, 0.0001, 0.00001
Inicialização dos pesos	Uniform, Normal, Glorot Normal, Glorot Uniform
Taxa de <i>dropout</i>	0.0, 0.1, 0.2, 0.3, 0.4, 0.5
Taxa de restrição dos pesos	1, 2, 3, 4, 5
Regularização	L1, L2, L1_L2

A biblioteca *scikit-optimize*² foi utilizada para realizar a busca de um conjunto de hiperparâmetros otimizado. Esse processo possui 50 iterações e uma validação cruzada de 10 divisões é aplicada em cada uma das repetições. Para verificar a qualidade das predições realizadas foram extraídas as 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*.

III. RESULTADOS E DISCUSSÃO

O primeiro passo realizado no trabalho foi a leitura das 108 planilhas do *Microsoft Excel*, contendo os dados de todos os experimentos executados. Com base nestes dados foi criado um

²<https://scikit-optimize.github.io/stable/>

Dataframe, um tipo de objeto da biblioteca *pandas*³. Este objeto contém 108 linhas e 11 colunas, representando os valores dos dados coletados durante os experimentos, incluindo a matriz 16x16 contendo os valores de precipitação dos pluviômetros em cada um dos experimentos.

Na sequência, a variável dependente contendo os valores de precipitação foi retirada do objeto Dataframe e ambos foram separados em conjuntos de treinamento, validação e teste. Os 10 arquivos criados, apresentados na Tabela IV, foram salvos para serem utilizados em todos os experimentos realizados.

TABELA IV
CONJUNTOS DE DADOS DESENVOLVIDOS

Dados de Treinamento	Dados de Validação	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

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 validação.

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. A Tabela V apresenta o conjunto de hiperparâmetros definidos pela busca Bayesiana para esse modelo.

TABELA V
ARQUITETURA DO MODELO QUE ALCANÇOU OS MELHORES RESULTADOS

Hiperparâmetro	80/20(10)
Algoritmo de Aprendizado	Adagrad
Taxa de Aprendizagem	0,1
Função de Ativação	Tanh
Modo de Inicialização	Normal
Taxa de Abandono	0,1
Taxa de Restrição	3
Número de Neurônios	1024
Função de Regularização	L1
Tamanho do mini-lote	6
Número de Épocas	500

Esse modelo alcançou um MSE de 50,6261 e um R^2 de 92,87%. Os modelos desenvolvidos tinham como objetivo

³<https://pandas.pydata.org/>

estimar os valores de precipitação em cada um dos 256 pluviômetros. Para facilitar a visualização dos resultados, o erro foi calculado com base no valor total de água observado em cada uma das repetições da validação cruzada.

Pela análise de comparação dos erros, apresentada na Figura 1, é possível observar 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.

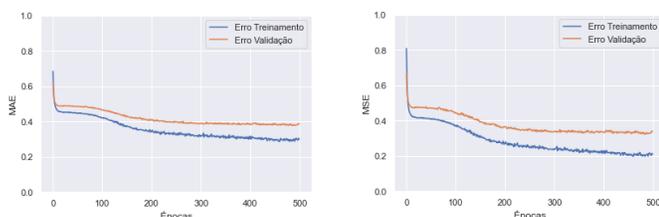


Fig. 1. MAE e MSE para os conjuntos de treinamento e validação do melhor modelo

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) foi calculado para as distâncias, em metros, de 12x12, 12x15 e 15x15.

Este processo foi realizado para verificar se o modelo selecionado é capaz de realizar boas estimativas dessa variável para diferentes distâncias entre aspersores. A Tabela VI apresenta os resultados desse processo.

TABELA VI
ARQUITETURA DO MODELO QUE ALCANÇOU OS MELHORES RESULTADOS

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

A Tabela VI apresenta os valores de CUC observados para cada uma das instâncias que formam o conjunto de dados de teste. Os melhores resultados alcançados foram na predição do CUC na distância 12x12, com o modelo selecionado tendo um erro médio de 0,79 para as instâncias apresentadas e um desvio padrão de 0,52.

Utilizando os conjuntos de hiperparâmetros que alcançaram os melhores resultados no modelo de uma camada, 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 VII apresenta os resultados desse processo.

TABELA VII
AVALIAÇÃO DOS RESULTADOS ALCANÇADOS PELO MODELO DE MÚLTIPLAS CAMADAS

Divisão dos Dados	Núm. de Camadas	Vars.	MAE	MSE	RMSE	R ²
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

A análise da Tabela VII indica que a utilização de um modelo com múltiplas camadas não trouxe melhoras para os resultados alcançados. A configuração de divisão dos dados na proporção de 70/30, com 10 variáveis de entrada e três camadas ocultas alcançou um RMSE de 65,01 e um R² de 91,03%, valores são inferiores aos alcançados pelo modelo com apenas uma camada.

[16] utilizou uma abordagem teórico-conceitual na aplicação de redes neurais artificiais para estimar a precipitação de aspersores. Utilizando diferentes algoritmos de aprendizagem, um modelo capaz de estimar a precipitação de água de um microaspersor foi criado. Esse estudo alcançou bons resultados estatísticos, com um coeficiente de determinação (R²) igual a 0,93.

Em sua pesquisa para simular a precipitação de microaspersores com a ajuda de redes neurais artificiais, [9] 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.

O melhor modelo analisado neste trabalho utilizou 10 variáveis de entrada. Porém, a utilização de mais variáveis pode acarretar em um aumento na dificuldade de execução dos experimentos. Como os modelos com cinco variáveis de entrada alcançaram valores próximos de R² em relação ao melhor modelo analisado, que utilizou 10 variáveis de entrada e alcançou um MSE de 50,62 e um R² de 92,87%, a utilização de menos variáveis onde a coleta de dados é mais fácil pode ser considerada uma escolha válida.

IV. CONCLUSÃO

Este trabalho se propôs a realizar o desenvolvimento de modelos de redes neurais capazes de prever a quantidade de água utilizada por microaspersores e estimar corretamente seu coeficiente de uniformidade. 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 informações meteorológicas e de precipitação, para o conjunto de treinamento e 20% para o conjunto de validação.

Além disso foram utilizadas 10 variáveis de entrada. Esta configuração alcançou um MSE de 50,62 e um R^2 de 92,87%. 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 previsões de CUC. Utilizando o conjunto de dados de teste, composto 7% dos dados originais e que foram escolhidos aleatoriamente antes da divisão dos conjuntos de treinamento e validação, o modelo teve um erro médio de 0,79 nos valores de CUC utilizando a distância de 12x12m.

Como possibilidade de novas pesquisas que possam contribuir para o enriquecimento deste trabalho podem ser citadas a utilização de um conjunto de dados maior para verificar se esse fator irá contribuir para uma mudança nos resultados e a verificação da capacidade do modelo desenvolvido em estimar outros coeficientes de uniformidade, como o Coeficiente de Uniformidade de Distribuição (CUD) e o Coeficiente de Uniformidade Estatístico (CUE).

- [10] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020.
- [11] I. Shcherbatyi, T. Head, and G. Louppe. (2017) Scikit-learn hyperparameter search wrapper. [Online]. Available: https://scikit-optimize.github.io/stable/auto_examples/sklearn-gridsearchcv-replacement.html
- [12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [13] E. Jumin, F. B. Basaruddin, Y. B. Yusoff, S. D. Latif, A. N. Ahmed *et al.*, "Solar radiation prediction using boosted decision tree regression model: A case study in malaysia," *Environmental Science and Pollution Research*, vol. 28, no. 21, pp. 26 571–26 583, 2021.
- [14] A. Wuraola, N. Patel, and S. K. Nguang, "Efficient activation functions for embedded inference engines," *Neurocomputing*, vol. 442, pp. 73–88, 2021.
- [15] C. K. Dewa *et al.*, "Suitable cnn weight initialization and activation function for javanese vowels classification," *Procedia computer science*, vol. 144, pp. 124–132, 2018.
- [16] J. Wolfrann *et al.*, "Redes neurais artificiais para estimar a precipitação na irrigação por aspersão," Master's thesis, Universidade Tecnológica Federal do Paraná, 2018.

REFERÊNCIAS

- [1] S. Pandey, "The role of irrigation for food security and sustainability," in *Encyclopedia of Food Security and Sustainability*, P. Ferranti, E. M. Berry, and J. R. Anderson, Eds. Oxford: Elsevier, 2019, pp. 142–146. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780081005965221732>
- [2] R. Testezlaf, "Irrigação: métodos, sistemas e aplicações," 2011.
- [3] A. Phocaidis, *Handbook on pressurized irrigation techniques*. Food & Agriculture Org., 2007.
- [4] O. Sheikhesmaeili, J. Montero, and S. Laserna, "Analysis of water application with semi-portable big size sprinkler irrigation systems in semi-arid areas," *Agricultural Water Management*, vol. 163, pp. 275–284, 2016.
- [5] B. Salvatierra-Bellido, J. Montero-Martínez, and L. Pérez-Urrestarazu, "Development of an automatic test bench to assess sprinkler irrigation uniformity in different wind conditions," *Computers and Electronics in Agriculture*, vol. 151, pp. 31–40, 2018.
- [6] P. Carrión, J. Tarjuelo, and J. Montero, "Sirias: a simulation model for sprinkler irrigation," *Irrigation Science*, vol. 20, no. 2, pp. 73–84, 2001.
- [7] S. Sayari, A. Mahdavi-Meymand, and M. Zounemat-Kermani, "Irrigation water infiltration modeling using machine learning," *Computers and Electronics in Agriculture*, vol. 180, p. 105921, 2021.
- [8] A. Dubois, F. Teytaud, and S. Verel, "Short term soil moisture forecasts for potato crop farming: A machine learning approach," *Computers and Electronics in Agriculture*, vol. 180, p. 105902, 2021.
- [9] P. L. d. Menezes, C. A. de Azevedo, E. Eyng, J. Dantas Neto, and V. L. de Lima, "Artificial neural network model for simulation of water distribution in sprinkle irrigation," *Revista Brasileira de Engenharia Agrícola e Ambiental*, vol. 19, pp. 817–822, 2015.