Aplicação de Técnicas de Aprendizado de Máquina para Classificação de Isquemia e Infecção em Úlcera do Pé Diabético

Lívia Marques Rodrigues
Departamento de Automática
Universidade Federal de Lavras (UFLA)
Lavras, Brasil
livia.rodrigues3@estudante.ufla.br

Priscila Peruzzo Apolinario
Faculdade de Enfermagem
Universidade Estadual de Campinas (UNICAMP)
Campinas, Brasil
priscilapolinario@gmail.com

Bruno da Silva Macêdo
Departamento de Automática
Universidade Federal de Lavras (UFLA)
Lavras, Brasil
bruno.macedo2@estudante.ufla.br

Júlio César Sousa Lima
Departamento de Automática
Universidade Federal de Lavras (UFLA)
Lavras, Brasil
julio.lima1@estudante.ufla.br

Ana Cláudia B. Honório Ferreira
Enfermagem
Centro Universitário Lavras (UNILAVRAS)
Lavras, Brasil
ananepe@yahoo.com.br

Maria Helena de Melo Lima
Faculdade de Enfermagem
Universidade Estadual de Campinas (UNICAMP)
Campinas, Brasil
melolima@unicamp.br

Danton Diego Ferreira
Departamento de Automática
Universidade Federal de Lavras (UFLA)
Lavras, Brasil
danton@ufla.br

***Resumo*—A classificação de isquemia e infecção em úlceras do pé diabético é um desafio clínico com impacto direto na prevenção de amputações. Este estudo propõe um sistema inteligente baseado em classificadores tradicionais, com foco na sustentabilidade e baixo custo computacional. Foram comparados métodos como *One Class Principal Curves* (OCPC), *Random Forest* (RF), *Gradient Boosting* e *Multilayer Perceptron* (MLP), com e sem o pré-processamento via Análise de Componentes Principais (PCA), avaliando não apenas métricas de desempenho, mas também tempo de predição e pegada de carbono (CO2eq) durante o treinamento e durante a predição de um indivíduo. O melhor resultado foi alcançado pelo *XGBoost*, com acurácia de 0.92 na classificação de isquemia e 0.75 na de infecção, superando a abordagem baseada em aprendizado profundo, tanto em desempenho quanto em custo computacional, demonstrando que abordagens mais simples podem oferecer soluções competitivas com aquelas oferecidas por modelos de aprendizado profundo, e viáveis para aplicações em larga escala, especialmente em contextos de recursos limitados.**

***Palavras-chave*—Inteligência Artificial, *Machine Learning*, Pegada de Carbono, Sustentabilidade, Baixo Custo Computacional.**

I. Introdução

Úlceras do Pé Diabético (UPD) são uma complicação grave do diabetes, e o seu tratamento representa um grande problema global de saúde, resultando em altos custos de cuidados e taxas elevadas de mortalidade. O reconhecimento de infecção e isquemia, associada à Doença Arterial Periférica (DAP), é essencial para determinar os fatores que influenciam o progresso da cicatrização das UPD e o risco de amputação. A isquemia é causada principalmente pela DAP, e resulta da obstrução ou estreitamento das artérias dos membros inferiores, levando à redução do fluxo sanguíneo e, consequentemente, à diminuição da perfusão tecidual. Isso pode resultar em gangrena na úlcera, o que pode exigir a amputação de parte do membro se não for reconhecida e tratada precocemente. Um conhecimento detalhado da anatomia vascular da perna, especialmente da isquemia, permite que profissionais de saúde tomem melhores decisões ao estimar a possibilidade de cicatrização da UPD, considerando o suprimento sanguíneo existente [1].

Este artigo foca na identificação da isquemia e infecção de UPD, que são definidas da seguinte forma [2], [3]:

1) **Isquemia**: Suprimento sanguíneo inadequado que pode afetar a cicatrização da UPD. A isquemia é diagnosticada através da avaliação de perfusão, que é fundamental, e deve incluir exames como o Índice Tornozelo-Braquial (ITB), Índice Dedo-Braquial (IDB) e medidas transcutâneas de oxigênio. Visualmente, a isquemia pode ser indicada pela má reperfusão no pé ou pela presença de dedos gangrenados (morte tecidual em parte do pé). Do ponto de vista da visão computacional, esses

Agradecimentos a Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) e Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

sinais podem ser informações importantes da presença de isquemia na UPD.

2) **Infecção Bacteriana**: A infecção bacteriana de tecido mole ou ósseo na UPD, baseada na presença de ao menos dois sinais clássicos de inflamação ou exsudato purulento. É difícil determinar a presença de infecções a partir de imagens de UPD, mas hiperemia acima de 2 cm ao redor da úlcera e presença de exsudato purulento podem ser indicativos. Além disso, no conjunto de dados utilizado neste estudo, as imagens foram capturadas após desbridamento dos tecidos necróticos e desvitalizados, o que remove uma indicação importante dos sinais de infecção. Assim, fica claro que analisar essas condições por imagem é uma tarefa difícil para os profissionais de saúde.

Sabe-se que, em várias aplicações de reconhecimento de imagens médicas e tarefas de processamento de linguagem natural, algoritmos de *Machine Learning* (ML) superaram humanos qualificados, inclusive médicos [3], [4], [5]. O cenário atual em diagnóstico auxiliado por imagem para UPD é composto por várias etapas: pré-processamento de imagem, segmentação, extração de características e classificação. Veredas et al. [6] propuseram o uso de características de cor e textura da área segmentada e uma rede neural multicamada para classificar tecidos e prever a cicatrização. Wannous et al. [7] classificaram tecidos a partir de descritores de cor e textura em um modelo 3D da ferida. Wang et al. [8] utilizaram um classificador em cascata de duas etapas para determinar os limites da UPD e sua área. Avanços importantes no ML baseado em imagem, especialmente com algoritmos de *Deep Learning* (DL), permitem o uso extensivo de dados de imagem médica com modelos de ponta para oferecer melhores diagnósticos, tratamentos e previsões de doenças [9], [10]. Modelos de DL para UPD, apesar dos bons resultados, frequentemente demandam recursos computacionais significativos, o que limita sua aplicabilidade em ambientes com restrições de infraestrutura. Além disso, esses modelos têm uma alta pegada de carbono, contribuindo negativamente para o impacto ambiental da computação em saúde [11], [12], [13], [14]. Tais fatores tornam urgente o desenvolvimento de abordagens que aliem eficiência diagnóstica e sustentabilidade ambiental por meio de soluções de baixo custo computacional, o que caracteriza o principal objetivo deste trabalho.

Os principais problemas e desafios na avaliação de UPD usando ML com imagens dos pés incluem: (i) o grande esforço necessário para coleta de dados e rotulagem por especialistas, além de uma alta semelhança entre classes diferentes e variações dentro da mesma classe dependendo da classificação da UPD; e (ii) a falta de padronização no conjunto de dados, como distância da câmera ao pé, orientação da imagem e condições de iluminação, e a ausência de metadados, como etnia, idade, sexo e tamanho do pé do paciente.

Especialistas na área de úlceras no pé de pessoas com diabetes têm boa experiência em prever a presença de isquemia ou infecção a partir da análise da úlcera. O objetivo deste trabalho é automatizar este processo via ML. Para aumentar a confiabilidade da rotulagem, quatro especialistas previram a presença de isquemia e infecção a partir das imagens das UPD no conjunto de dados utilizado. Neste contexto, este artigo propõe um modelo de ML com baixo custo computacional e baixa pegada de carbono para auxiliar no reconhecimento de isquemia e infecção em UPD a partir de imagens. A proposta visa equilibrar eficiência diagnóstica e sustentabilidade, sendo adequada para contextos clínicos com recursos computacionais limitados. Ao empregar algoritmos mais leves e eficientes, busca-se contribuir com soluções escaláveis e ambientalmente responsáveis para a triagem e acompanhamento de pacientes com UPD. O diferencial deste trabalho reside na avaliação integrada de desempenho preditivo e impacto ambiental de classificadores clássicos otimizados, com resultados superiores a métodos de DL em determinados cenários, especialmente sob restrições computacionais.

II. Materiais e Métodos

Nesta seção, são detalhados o conjunto de dados utilizado, o processo de pré-processamento aplicado e uma explicação sobre o conceito de pegada de carbono.

A. Base de Dados

Neste estudo, foi utilizado o conjunto de dados públicos desenvolvido por Goyal et al. [15], que contém imagens clínicas de UPD rotuladas quanto à presença ou ausência de isquemia e infecção. O banco de dados está organizado em duas abordagens distintas, cada uma tratando de um aspecto clínico específico: uma voltada para a detecção de isquemia e outra voltada para a detecção de infecção. Portanto, podem ser considerados dois subconjuntos separados de dados, cada um com seu próprio conjunto de imagens e rótulos associados.

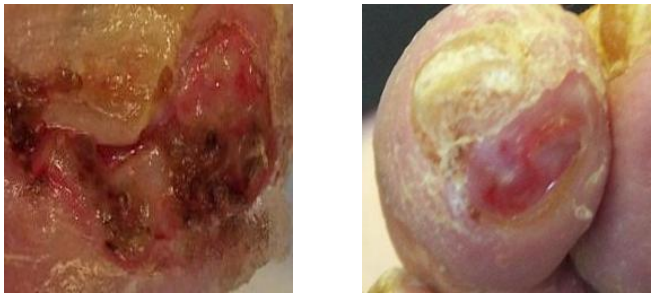
O *dataset* referente à presença ou ausência de isquemia é composto por 4.935 imagens da classe “isquemia” e 4.935 imagens da classe “não isquemia”. Já o subconjunto relacionado à detecção de infecção contém 2.946 imagens rotuladas como “infecção” e 2.946 imagens como “sem infecção”. Esses números incluem tanto as imagens originais quanto suas versões aumentadas por transformações geométricas, com o objetivo de balancear as classes e enriquecer a variabilidade do conjunto de treinamento.

Cada imagem é nomeada seguindo uma convenção que permite identificar sua origem e transformação. Imagens com final “_1X” representam as versões originais, enquanto “_2X” e “_3X” indicam versões aumentadas por transformações geométricas. Além disso, alguns arquivos incluem sufixos como “M” (espelhamento horizontal) e “R1”, “R2” e “R4”, correspondentes a rotações de 90°, 180° e 270°, respectivamente. As Figs. 1 e 2 ilustram exemplos visuais desses dados, representando as quatro condições avaliadas: presença e ausência de isquemia, e presença e ausência de infecção.

O diferencial metodológico importante desse conjunto de dados é a utilização de uma estratégia unificada de aumento de dados, que visa expandir o número de amostras disponíveis e melhorar a robustez dos modelos de ML aplicados, sem necessidade de coletar novos dados reais. As transformações

(a) (b)

Fig. 1. Exemplos de imagens do conjunto de dados: (a) presença de isquemia, (b) ausência de isquemia. Fonte: Goyal et al. [15].



(a) (b)

Fig. 2. Exemplos de imagens do conjunto de dados: (a) presença de infecção, (b) ausência de infecção. Fonte: Goyal et al. [15].

usadas são simples, mas eficazes, e foram aplicadas de forma padronizada pelos autores do *dataset*.

Essa estratégia de aumento de dados é especialmente relevante para contextos de saúde, onde a coleta de imagens rotuladas por especialistas é cara e demorada. O seu uso contribui para a diversidade dos dados e ajuda a reduzir o risco de *overfitting* durante o treinamento dos modelos.

B. Pré-Processamento

As imagens do conjunto de dados, que possuem 64x64 *pixels*, foram normalizadas, convertendo os valores dos *pixels* de inteiros no intervalo $[0,\ 255]$ para valores de ponto flutuante no intervalo entre 0 e 1. Isso foi feito dividindo cada valor de *pixel* por 255. A normalização é uma técnica fundamental para melhorar o desempenho e a estabilidade dos algoritmos, pois evita que atributos com maiores magnitudes dominem os demais, além de acelerar a convergência dos métodos [16].

Em seguida, cada imagem foi convertida em um vetor unidimensional por meio do processo de (*flatten*), convertendo o formato de matriz tridimensional (altura, largura, canais) para um vetor de características unidimensional. Essa transformação permite que as imagens sejam processadas por modelos clássicos de ML que exigem entradas vetoriais ao invés de matrizes [17].

Para evitar vazamento de informação entre os conjuntos de treino e teste, a matriz de transformação PCA foi ajustada exclusivamente com os dados de treinamento e posteriormente aplicada aos dados de teste. Essa abordagem simula de forma mais fiel o uso real do modelo em produção, além de permitir a avaliação correta da escalabilidade e do custo computacional. O vetor original das imagens (64×64×3) resultou em vetores unidimensionais com 12.288 atributos, reduzidos para 50 componentes principais, preservando a maior parte da variância explicada.

C. Pegada de Carbono

Para avaliar o impacto ambiental do treinamento dos modelos de ML utilizados neste estudo, foi empregada a biblioteca *codecarbon* [18], uma ferramenta em *Python* que monitora e estima as emissões associadas à execução dos experimentos. O *codecarbon* realiza o rastreamento do consumo energético em tempo real durante o processo de treinamento, levando em consideração informações sobre o *hardware* utilizado (como CPU e GPU) e a localização geográfica, a fim de fornecer uma estimativa precisa da pegada de carbono gerada.

As emissões foram medidas em CO_2 equivalente (CO_2eq), unidade que representa o impacto climático de diferentes gases de efeito estufa convertidos para o equivalente ao dióxido de carbono, o utilizado como padrão internacional para quantificar a pegada de carbono de processos computacionais.

Essa métrica ambiental foi coletada para todos os modelos avaliados no conjunto de teste, com o objetivo de quantificar e comparar a eficiência energética dos algoritmos. A inclusão dessa análise possibilita uma avaliação mais abrangente, considerando não apenas o desempenho preditivo dos modelos, mas também seu custo ambiental, destacando a sustentabilidade da abordagem adotada.

O uso de métricas ambientais em experimentos de ML tem ganhado atenção nos últimos anos, diante da crescente demanda computacional da área. Estudos como o de Strubell et al. [19] demonstram que o treinamento de modelos de grande porte pode emitir quantidades significativas de CO_2, o que reforça a importância de se considerar o impacto ambiental como parte integrante das boas práticas científicas. Nesse sentido, ferramentas como o *codecarbon* promovem maior transparência e responsabilidade no desenvolvimento de soluções baseadas em Inteligência Artificial (IA) [20], [21].

III. Métodos de Classificação

Nesta seção, são apresentados os métodos de classificação utilizados. Para os classificadores *Random Forest (RF)*, *Extreme Gradient Boosting* (XGBOOST) e *Multilayer Perceptron* (MLP), foi aplicada uma redução de dimensionalidade via Análise de Componentes Principais (PCA) [22], com retenção das 50 principais componentes, antes do treinamento e validação dos modelos. Essa abordagem foi escolhida para diminuir a complexidade computacional, ao mesmo tempo, em que preserva as características mais relevantes dos dados, descartando as informações redundantes.

Além disso, nesta seção também são apresentadas as métricas de desempenho utilizadas para verificar o desempenho dos classificadores. As métricas incluem acurácia, precisão, *recall*, *F1-score* e AUC. Ademais, com o objetivo de desenvolver soluções eficientes e sustentáveis, foram mensuradas a pegada de carbono associada ao treinamento final dos modelos e o tempo médio de inferência por imagem.

A. *One-Class Classifier based on Principal Curves (OCPC)*

O algoritmo *One-Class Classifier based on Principal Curves* (OCPC) é uma técnica baseada em curvas principais, proposta por Borges et al. [23], que visa representar a distribuição de dados de forma não linear, explorando a geometria intrínseca dos conjuntos de dados. As curvas principais estendem o conceito de análise de componentes principais (PCA), permitindo que uma curva suave atravesse o "meio" da distribuição dos dados, capturando estruturas mais complexas e não lineares. Essa propriedade torna o OCPC especialmente adequado para dados que apresentam formas alongadas, ramificações ou trajetórias em espaços de alta dimensionalidade, onde métodos lineares falham em capturar a verdadeira estrutura da informação.

Na abordagem multiclasse, utilizada neste trabalho, o algoritmo ajusta uma curva principal distinta para cada classe presente no conjunto de dados. Durante o processo de inferência, uma nova amostra é projetada sobre todas as curvas principais aprendidas, sendo atribuída à classe cuja curva correspondente apresenta a menor distância ortogonal em relação à amostra. Esse mecanismo de decisão permite preservar a capacidade do OCPC de modelar a forma de cada classe de forma independente, resultando em uma classificação precisa mesmo em contextos onde as classes possuem distribuições altamente não lineares.

Além de sua flexibilidade geométrica, o OCPC se destaca também pela sua eficiência computacional na fase de inferência, já que a classificação é realizada com base em cálculos diretos de projeção e distância. Os experimentos apresentados por Borges et al. [23] demonstram que, mesmo quando comparado com métodos clássicos e modernos de aprendizado supervisionado, o OCPC multiclasse é capaz de construir fronteiras de decisão suaves e bem definidas, o que contribui para sua robustez e capacidade de generalização. Dessa forma, sua aplicação em cenários com múltiplas classes se mostra uma alternativa eficaz e interpretável, particularmente vantajosa quando se deseja capturar a morfologia real das distribuições das classes em espaços complexos.

B. *Random Forest (RF)*

Random Forest (RF), proposto por Breiman [24], é um método de aprendizagem por *ensemble* que opera construindo p árvores de decisão a partir do conjunto de treinamento em p iterações. Em cada iteração, primeiro é selecionado aleatoriamente um conjunto de amostras do conjunto de treinamento. Para reproduzir uma árvore de decisão desse subconjunto, o RF escolhe aleatoriamente um subconjunto de atributos candidatos para cada nó. Desta forma, cada árvore de decisão é construída através do *ensemble*, empregando subconjuntos aleatórios independentes de características e amostras. A predição de uma nova classe de amostra é realizada da seguinte forma: cada classificador individual vota e a classe mais votada é eleita.

A Fig. 3 ilustra um exemplo do algoritmo RF, na qual ocorre inicialmente a entrada de uma instância. Em seguida, várias árvores de decisão são geradas (Árvore 1, Árvore 2, etc), sendo cada uma responsável por atribuir uma classe à instância, como Classe A ou Classe B. Após cada árvore realizar sua previsão, o resultado é determinado por meio de uma votação máxima, onde a classe que receber o maior número de votos entre as árvores é escolhida como a classificação final.

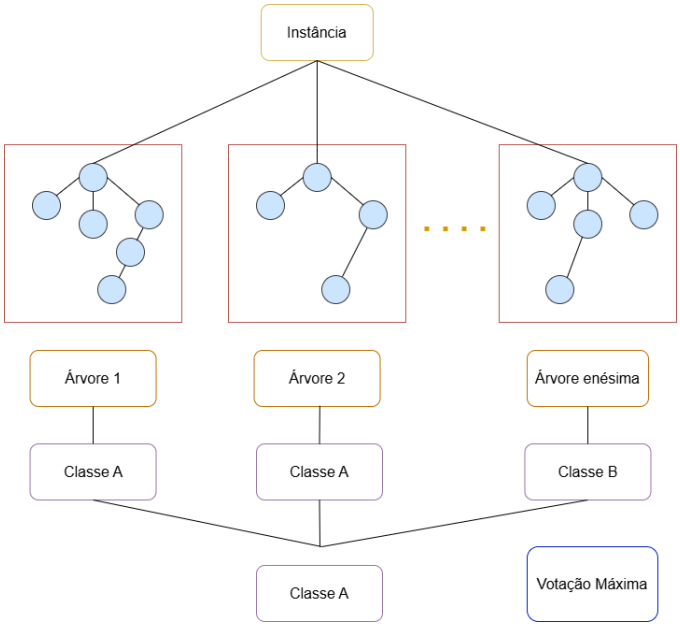


Fig. 3. Exemplo de uma arquitetura do RF. Fonte: Elaborado pelos autores.

Neste trabalho, foi utilizada uma floresta composta por 300 árvores, com profundidade máxima de 20, amostragem balanceada entre classes e paralelização ativada para acelerar o treinamento.

C. *Extreme Gradient Boosting (XGBOOST)*

O algoritmo *Extreme Gradient Boosting* (XGBOOST) é um método adaptado do *Gradiente Boosting* que demonstrou ser mais eficaz para resolver problemas de aprendizagem supervisionada [25]. Algumas etapas são executadas, como adotar um conjunto de dados com um número n de amostras $(x_1, y_1), ..., (x_n, y_n)$ onde $x_i \in \mathbb{R}^n$ e $y_i \in \mathbb{R}$, $i = 0, ...n$. Adotando-se $\hat{y}_i$ como a saída prevista pelo modelo XGBOOST para a amostra x_i, tem-se:

$$\phi(x_i) = \sum_{k=1}^{K} h_k(x_i), \qquad h_k \in \mathbb{H} \tag{1}$$

em que K representa o número de árvores de decisão, h_k é o modelo da k-ésima árvore de decisão com profundidade máxima definida para o modelo h_k. Na geração do modelo, é necessário minimizar a função de perda de regularização

$$L(\phi) = \sum_{j} l(y_j, \phi(x_j)) + \frac{1}{2}\left(\sum_{j} w_j^2\right)^{\frac{1}{2}} \tag{2}$$

em que a função de perda é $l = \|\hat{y}_i - y_i\|$, $\hat{y}_i$ e y_i são a saída estimada e verdadeira, respectivamente, e $w = (w_1, w_2, \dots)$ é o vetor peso da folha.

Para este trabalho, foi utilizado um classificador *Gradient Boosting* com 300 árvores, profundidade máxima de 8, taxa de aprendizado de 0.1, balanceamento entre classes e paralelização ativada para acelerar o treinamento.

D. Multilayer Perceptron (MLP)

As redes neurais *Multilayer Perceptron* (MLP) representam um paradigma de programação inspirado na estrutura do cérebro [26]. Elas foram desenvolvidas com o intuito de resolver problemas que são não linearmente separáveis em um plano. A estrutura do MLP é composta por três componentes principais: uma camada de entrada, na qual ocorre a inserção dos dados, uma ou mais camadas ocultas para o processamento das informações e aprendizado da rede, e uma camada de saída da rede, em que é retornada a variável de saída. Um neurônio *K* é definido pelas Eqs. (3) e (4).

$$y_k = f(u_k + b_k) \tag{3}$$

$$u_k = \sum_{i=1}^{N} W_{ki} X_i \tag{4}$$

onde, X_i representa o dado de entrada, W_{ki} o peso de conexão do neurônio, u_k a saída linear, b_k é o termo de polarização, f é a função de ativação e y_k é o valor de saída do neurônio.

Foi utilizado neste trabalho um MLP com duas camadas ocultas contendo 128 e 64 neurônios, respectivamente, função de ativação *ReLU*, otimizador *Adaptive Moment Estimation* (Adam) [27].

E. Validação Cruzada (VC)

A Validação Cruzada (VC) é uma técnica para avaliar a capacidade de generalização de um modelo a partir do conjunto de dados. Essa abordagem é muito aplicada em problemas onde o objetivo é realizar a predição. A VC divide o banco de dados em dois conjuntos distintos: treinamento e teste. O conjunto de treinamento é usado para estimar os parâmetros do modelo e o conjunto de teste é usado para validar o modelo.

Para avaliar os classificadores, foi empregada a estratégia *K-Fold* (KF) [28]. A estratégia KF no banco de dados disponível contém N amostras e é dividida em L subconjuntos, onde $L > 1$. Após particionar o banco de dados, os subconjuntos gerados por L-1 são usados para treinamento e os conjuntos restantes são utilizados para teste. Desta forma, ao final do procedimento, mede-se o erro de validação. Este processo é repetido L vezes, cada vez usando um conjunto de teste diferente em cada iteração. O desempenho do classificador é calculado em L testes. O objetivo de repetir os testes várias vezes é treinar da melhor forma possível a máquina para que ela possa generalizar entradas futuras. Neste trabalho, foi utilizado $L = 5$.

F. Métricas de Avaliação

Foram empregadas as seguintes métricas para avaliar o desempenho dos algoritmos: Acurácia, Precisão, *Recall*, *F1-Score* e Área Sob a Curva (AUC). Nas equações dessas métricas, apresentadas a seguir, *True Positive* (TP) corresponde ao número de amostras corretamente classificadas como pertencentes a uma classe específica. Já os *False Positive* (FP) indicam a quantidade de amostras incorretamente classificadas como pertencentes a essa classe. Por sua vez, *True Negative* (TN) representa o número de amostras erroneamente identificadas como não pertencentes à classe em análise.

A acurácia é uma métrica que indica a proporção de previsões corretas realizadas pelo modelo, considerando tanto os TP quanto os TN, em relação ao total de previsões. Sua fórmula é dada por:

$$\text{Acurácia} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

A precisão mede a proporção de amostras classificadas como positivas de fato positivas. Essa métrica é especialmente útil quando o custo de FP é alto:

$$\text{Precisão} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

O *Recall*, ou sensibilidade, avalia a capacidade do modelo em identificar corretamente todas as amostras positivas, sendo particularmente relevante quando o custo de falsos negativos é alto:

$$\textit{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

O *F1-Score* é a média harmônica entre precisão e *Recall*. Ele é especialmente útil em contextos com classes desbalanceadas, pois combina essas duas métricas em uma única medida:

$$\textit{F1-Score} = 2 \cdot \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}} \tag{8}$$

A Área Sob a Curva ROC (*Area Under the Curve* (AUC)) é uma métrica que avalia a capacidade do modelo de diferenciar entre as classes positiva e negativa em diversos limiares de decisão. A curva ROC é elaborada com base na relação entre a taxa de verdadeiros positivos (sensibilidade) e a taxa de falsos positivos. O valor da AUC varia de 0 a 1, sendo que, quanto mais próximo de 1, melhor o desempenho do modelo, enquanto valores em torno de 0.5 indicam um desempenho equivalente ao acaso.

IV. Resultados e Discussões

Os experimentos computacionais foram conduzidos utilizando a linguagem de programação *Python*, com implementações baseadas na biblioteca *codecarbon* [18], *ocpc_py* [29] e *Scikit-Learn* [30]. Todos os testes foram executados em um computador com processador *AMD Ryzen 5 7535HS*, 8 GB de RAM e sistema operacional *Windows 11*.

Para a vetorização das imagens, utilizaram-se as funções *resize* e *flatten* da biblioteca *OpenCV* e *NumPy*, respectivamente, para transformação em vetores unidimensionais. A redução de dimensionalidade foi realizada com a função PCA da biblioteca *decomposition*, do pacote *Scikit-Learn*. Já a biblioteca *ocpc-py* foi utilizada para aplicação do classificador OCPC.

Os resultados obtidos revelam um equilíbrio relevante entre desempenho preditivo e custo computacional entre os classificadores avaliados. As análises envolveram métricas clássicas de desempenho (acurácia, precisão, *Recall*, *F1-score* e AUC), bem como indicadores de sustentabilidade computacional, como pegada de carbono e tempo médio de inferência por imagem.

A. Classificação de Isquemia

Na tarefa de classificação de isquemia, como mostrado nas Tabelas I e II, o classificador *Gradient Boosting* sem aplicação de PCA destacou-se como o mais eficaz, alcançando acurácia de 0.9200, F1-score de 0.9200 e AUC de 0.9781 no conjunto de teste. Os desempenhos obtidos durante a VC são da mesma ordem (vide Tabela I). Contudo, esse alto desempenho resultou no maior custo ambiental observado, com pegada de carbono emitida no treinamento de 0.000282 kg CO_2eq.

TABELA I
MÉTRICAS MÉDIAS DURANTE A VALIDAÇÃO CRUZADA - ISQUEMIA

Classificador	Acurácia	Precisão	Recall	F1-Score	AUC
OCPC com PCA	0.6982 ± 0.0109	0.7226 ± 0.0143	0.6441 ± 0.0069	0.6811 ± 0.0091	0.7728 ± 0.0124
Random Forest com PCA	0.8649 ± 0.0070	0.8379 ± 0.0056	0.9048 ± 0.0125	0.8700 ± 0.0070	0.9415 ± 0.0043
Random Forest	0.8774 ± 0.0061	0.8663 ± 0.0082	0.8927 ± 0.0105	0.8792 ± 0.0060	0.9492 ± 0.0045
Gradient Boosting com PCA	0.8929 ± 0.0093	0.8836 ± 0.0055	0.9051 ± 0.0145	0.8942 ± 0.0087	0.9585 ± 0.0038
Gradient Boosting	**0.9158 ± 0.0052**	**0.9172 ± 0.0041**	**0.9142 ± 0.0093**	**0.9157 ± 0.0043**	**0.9712 ± 0.0035**
MLP com PCA	0.8658 ± 0.0042	0.8598 ± 0.0086	0.8741 ± 0.0059	0.8668 ± 0.0045	0.9381 ± 0.0044
MLP	0.8254 ± 0.0107	0.8421 ± 0.0136	0.8005 ± 0.0134	0.8208 ± 0.0129	0.9075 ± 0.0091
Ensemble (CNN)	0.9030 ± 0.0120	0.9180 ± 0.0190	0.9210 ± 0.0210	0.9020 ± 0.0140	0.9040 ± 0.0120

TABELA II
MÉTRICAS DURANTE A VALIDAÇÃO COM CONJUNTO DE TESTE - ISQUEMIA

Classificador	Acurácia	Precisão	Recall	F1-Score	AUC	Pegada de Carbono
OCPC com PCA	0.7102	0.7248	0.6778	0.7005	0.7738	0.000011 kg CO_2eq
Random Forest com PCA	0.8723	0.8406	0.9189	0.8780	0.9493	0.000009 kg CO_2eq
Random Forest	0.8906	0.8806	0.9037	0.8920	0.9594	0.000036 kg CO_2eq
Gradient Boosting com PCA	0.9063	0.8871	0.9311	0.9086	0.9691	0.000012 kg CO_2eq
Gradient Boosting	**0.9200**	**0.9191**	**0.9210**	**0.9200**	**0.9781**	**0.000282 kg CO_2eq**
MLP com PCA	0.8713	0.8887	0.8490	0.8684	0.9427	0.000010 kg CO_2eq
MLP	0.8029	0.8250	0.7690	0.7960	0.8877	0.000088 kg CO_2eq

Em contrapartida, o *Gradient Boosting* com PCA apresentou desempenho quase equivalente (Acurácia de 0.9063 , *F1-score* de 0.9086, AUC de 0.9691), com pegada de carbono emitida durante o treinamento mais de 20 vezes menor e inferência significativamente mais rápida (0.602 ms por imagem). Esses fatores tornam essa versão mais adequada para aplicações em tempo real e em contextos com recursos computacionais limitados. Além disso, esse resultado evidencia a eficiência do PCA em reduzir a dimensão dos dados e tornar o modelo mais simples.

A aplicação de PCA também trouxe ganhos importantes para o MLP, que teve melhoria substancial em todas as métricas após a redução de dimensionalidade. O modelo MLP com PCA obteve Acurácia de 0.8713, *F1-score* de 0.8684, AUC de 0.9427, e destacou-se por ser o mais eficiente em termos energéticos, com pegada de carbono no treinamento de apenas 0.000010 kg CO_2eq e tempo de inferência de 0.151 ms, o menor entre todos os modelos testados.

O classificador OCPC com PCA, embora extremamente leve (inferência em 1.634 ms e pegada de 0.000011 kg CO_2eq emitida durante o treinamento), teve o pior desempenho preditivo, com Acurácia de 0.7102, *F1-score* de 0.7005 e AUC de 0.7738, o que limita sua aplicabilidade prática em cenários clínicos que exigem maior confiabilidade.

Goyal et al. [15], ao utilizarem um *ensemble* de redes neurais convolucionais (CNNs) combinado com técnicas computacionalmente intensivas, como fusão multi-nível de características e otimização profunda de hiperparâmetros, apresentaram desempenho inferior aos resultados obtidos nesta análise comparativa, mesmo utilizando o mesmo conjunto de dados. Os autores reportaram uma acurácia de 0.903 e uma AUC de 0.904, valores inferiores aos alcançados pelo classificador *XGBoost* neste estudo. Além disso, essa abordagem exige maior custo computacional, o que destaca a eficiência dos classificadores tradicionais usados neste trabalho, que mantêm alto desempenho com menor complexidade e impacto ambiental.

B. Classificação de Infecção

Na tarefa de classificação de infecção, os resultados foram, de modo geral, mais modestos, refletindo a maior complexidade da detecção visual dessa condição (Tabelas III e IV). O *Gradient Boosting* sem PCA manteve a liderança, atingindo Acurácia de 0.7547, *F1-score* de 0.7422 e AUC de 0.8446, seguido de perto pelo RF sem PCA, com Acurácia de 0.7402 para o conjunto de teste. Os resultados durante a VC foram compatíveis. Entretanto, assim como na isquemia, foi o classificador que teve a maior emissão de carbono durante o treinamento final (0.000182 kg CO_2eq).

TABELA III
MÉTRICAS MÉDIAS DURANTE A VALIDAÇÃO CRUZADA - INFECÇÃO

Classificador	Acurácia	Precisão	Recall	F1-Score	AUC
OCPC com PCA	0.5450 ± 0.0136	0.5506 ± 0.0283	0.4943 ± 0.0207	0.5204 ± 0.0190	0.5663 ± 0.0231
Random Forest com PCA	0.6984 ± 0.0128	0.6869 ± 0.0152	0.7303 ± 0.0213	0.7076 ± 0.0116	0.7726 ± 0.0073
Random Forest	0.7362 ± 0.0122	0.7609 ± 0.0046	0.6892 ± 0.0259	0.7230 ± 0.0132	0.8138 ± 0.0099
Gradient Boosting com PCA	0.7173 ± 0.0070	0.7151 ± 0.0173	0.7235 ± 0.0193	0.7189 ± 0.0076	0.7885 ± 0.0099
Gradient Boosting	**0.7500 ± 0.0151**	**0.7701 ± 0.0099**	**0.7133 ± 0.0232**	**0.7404 ± 0.0137**	**0.8335 ± 0.0093**
MLP com PCA	0.6959 ± 0.0129	0.7086 ± 0.0182	0.6650 ± 0.0288	0.6858 ± 0.0194	0.7686 ± 0.0123
MLP	0.6070 ± 0.0181	0.6321 ± 0.0307	0.5100 ± 0.0611	0.5626 ± 0.0433	0.6460 ± 0.0250
Ensemble (CNN)	0.7270 ± 0.0250	0.7350 ± 0.0360	0.7440 ± 0.0500	0.7220 ± 0.0280	0.7310 ± 0.0230

TABELA IV
MÉTRICAS DURANTE A VALIDAÇÃO COM CONJUNTO DE TESTE - INFECÇÃO

Classificador	Acurácia	Precisão	Recall	F1-Score	AUC	Pegada de Carbono
OCPC com PCA	0.5526	0.5572	0.5127	0.5340	0.5849	0.000021 kg CO_2eq
Random Forest com PCA	0.7029	0.6918	0.7317	0.7112	0.7875	0.000008 kg CO_2eq
Random Forest	0.7402	0.7737	0.6791	0.7233	0.8198	0.000021 kg CO_2eq
Gradient Boosting com PCA	0.7080	0.7138	0.6944	0.7040	0.7930	0.000006 kg CO_2eq
Gradient Boosting	**0.7547**	**0.7820**	**0.7063**	**0.7422**	**0.8446**	**0.000182 kg CO_2eq**
MLP com PCA	0.6952	0.6997	0.6842	0.6918	0.7688	0.000004 kg CO_2eq
MLP	0.6316	0.7078	0.4482	0.5489	0.6717	0.000087 kg CO_2eq

A versão com PCA, novamente, demonstrou ser uma alternativa sustentável e eficiente. Apesar de uma leve redução de desempenho (Acurácia de 0.7080, *F1-score* de 0.7040, AUC de 0.7930), esse modelo alcançou inferência em apenas 0.609 ms e pegada de carbono de 0.000006 kg CO_2eq, sendo ideal para uso em dispositivos com baixa capacidade computacional.

Assim como na tarefa anterior, o MLP com PCA superou sua versão original em desempenho (*F1-score* de 0.6918 contra 0.5489) e eficiência energética. Já o OCPC com PCA, mesmo mantendo sua leveza computacional (inferência em 1,212 ms), não apresentou resultados satisfatórios (*F1-score* de 0.5340, AUC de 0.5849).

No cenário de classificação de infecção, Goyal et al. [15] também empregaram um *ensemble* de redes convolucionais aliado a abordagens computacionalmente custosas. Apesar da sofisticação do método, os melhores resultados obtidos, através de *Ensemble* (CNN) foram inferiores aos alcançados neste trabalho, com uma acurácia de 0.727, e uma AUC de 0.731. A elevada complexidade computacional da proposta evidencia, mais uma vez, a vantagem dos métodos apresentados neste estudo.

C. Análise da Eficiência Computacional dos Classificadores

Os resultados de tempo de inferência e pegada de carbono (Tabelas V e VI) evidenciam diferenças estruturais importantes entre os modelos. O RF apresentou os maiores tempos e emissões, mesmo com a aplicação de PCA. Isso se deve ao fato de o algoritmo operar com muitas árvores de decisão independentes, cada uma processando múltiplas regras em paralelo, o que eleva significativamente a carga computacional durante a inferência.

TABELA V
PEGADA DE CARBONO E TEMPO MÉDIO DE INFERÊNCIA POR IMAGEM - ISQUEMIA

Classificador	Pegada de Carbono	Tempo
OCPC com PCA	0.0000000030 kg CO_2eq	1.634 ms
Random Forest com PCA	0.0000000678 kg CO_2eq	43.336 ms
Random Forest	0.0000000602 kg CO_2eq	38.760 ms
Gradient Boosting com PCA	0.0000000012 kg CO_2eq	0.602 ms
Gradient Boosting	0.0000000014 kg CO_2eq	0.749 ms
MLP com PCA	0.0000000005 kg CO_2eq	0.151 ms
MLP	0.0000000010 kg CO_2eq	0.557 ms

TABELA VI
PEGADA DE CARBONO E TEMPO MÉDIO DE INFERÊNCIA POR IMAGEM - INFECÇÃO

Classificador	Pegada de Carbono	Tempo
OCPC com PCA	0.0000000020 kg CO_2eq	1.212 ms
Random Forest com PCA	0.0000000632 kg CO_2eq	40.384 ms
Random Forest	0.0000000628 kg CO_2eq	40.525 ms
Gradient Boosting com PCA	0.0000000012 kg CO_2eq	0.609 ms
Gradient Boosting	0.0000000014 kg CO_2eq	0.797 ms
MLP com PCA	0.0000000005 kg CO_2eq	0.157 ms
MLP	0.0000000010 kg CO_2eq	0.562 ms

Por outro lado, o *Gradient Boosting*, embora também seja baseado em árvores, mostrou-se mais eficiente devido à sua implementação otimizada e ao uso de técnicas como poda, paralelização e regularização. O uso de PCA nesse modelo trouxe ganhos adicionais, reduzindo ainda mais o tempo e as emissões.

Os modelos com MLP com PCA apresentaram os menores tempos e emissões, resultado da simplificação arquitetural promovida pela redução de dimensionalidade na entrada. Isso evidencia o impacto positivo da redução de dimensionalidade na performance energética de redes neurais. Demonstrando que não apenas o tipo de modelo, mas também as estratégias de pré-processamento, como o PCA, influenciam diretamente a viabilidade de implantação de sistemas inteligentes em cenários clínicos com restrições de recursos.

Para fins de comparação em relação ao custo computacional e à pegada de carbono, uma DL do tipo *ResNet50* foi treinada. Seu desempenho em métricas foi similar aos encontrados por Goyal et al. [15], e ao medir a sua pegada de carbono durante o treinamento, obteve-se o valor de 0.017697 kg CO_2eq para isquemia e 0.01924 kg CO_2eq para infecção. Comparando com o maior valor registrado neste trabalho — em ambos os casos, sem o uso de PCA e utilizando *Gradient Boosting* — obteve-se 0.000282 kg CO_2eq para isquemia e 0.000182 kg CO_2eq para infecção.

Esses resultados evidenciam que, embora a *ResNet50* alcance desempenhos similares em termos de acurácia e AUC, seu custo ambiental e computacional é substancialmente maior - aproximadamente 62 vezes mais emissões para isquemia e 105 vezes mais para infecção, quando comparada ao modelo mais custoso entre os métodos propostos neste trabalho. Essa diferença reforça a eficiência dos modelos baseados em técnicas de aprendizado de máquinas clássicas, como o *Gradient Boosting* aliado à redução de dimensionalidade via PCA. Assim, a proposta deste estudo demonstra não apenas competitividade em termos de desempenho preditivo, mas também vantagens consideráveis em termos de economia energética e redução de impacto ambiental.

D. Considerações Gerais

Os resultados demonstram que o *Gradient Boosting*, tanto com e sem o PCA, é o classificador mais robusto para classificação de isquemia e infecção em úlceras do pé diabético. Para contextos onde a máxima performance preditiva é desejada e os recursos computacionais não são limitantes, a versão sem PCA é a melhor escolha. Por outro lado, para aplicações que exigem baixo custo computacional e menor impacto ambiental, a versão com PCA oferece um excelente equilíbrio entre desempenho e sustentabilidade.

Além disso, os ganhos observados com a aplicação de PCA em classificadores como MLP e RF reforçam sua importância na otimização de modelos clássicos, possibilitando o desenvolvimento de soluções mais leves, rápidas e ambientalmente responsáveis. Essas soluções mostram-se particularmente relevantes para sistemas automatizados de triagem em ambientes clínicos com infraestrutura limitada, contribuindo para um diagnóstico mais acessível e sustentável.

V. CONCLUSÕES

Este estudo propôs um sistema inteligente de classificação de isquemia e infecção em UPD, utilizando algoritmos de ML com foco em baixo custo computacional e sustentabilidade ambiental. Os resultados experimentais demonstraram que, embora o *Gradient Boosting* sem PCA tenha alcançado o

melhor desempenho preditivo geral, ele apresentou também o maior custo ambiental em termos de pegada de carbono.

Por outro lado, versões otimizadas dos modelos com redução de dimensionalidade via PCA apresentaram desempenho competitivo, com vantagens significativas em termos de tempo de inferência e impacto ambiental. O modelo *Gradient Boosting* com PCA, em particular, destacou-se como uma alternativa balanceada entre acurácia e sustentabilidade, sendo indicado para aplicações em ambientes clínicos com infraestrutura computacional limitada. Além disso, os ganhos obtidos com a aplicação de PCA em modelos como MLP e RF evidenciam a importância de abordagens otimizadas para viabilizar soluções eficientes e escaláveis em contextos médicos reais. A inclusão da pegada de carbono como métrica de avaliação estabelece um novo paradigma, alinhando desempenho técnico com responsabilidade ambiental no desenvolvimento de soluções em IA.

Como trabalhos futuros serão exploradas estratégias adicionais de redução de consumo energético, como compressão de modelos ou *hardware* dedicado, bem como avaliar o sistema em dados clínicos reais e não aumentados, de modo a validar sua aplicabilidade prática em ambientes de saúde, além de trabalhos mais detalhados em termos de comparação com estratégias de aprendizado profundo.

AGRADECIMENTOS

Os autores agradecem o apoio financeiro da Universidade Federal de Lavras (UFLA), da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

REFERÊNCIAS

[1] IWGDF – International Working Group on the Diabetic Foot, "Iwgdf guidelines on the classification of diabetic foot ulcers 2023," https://iwgdfguidelines.org/guidelines/classification/, 2023, the Hague: IWGDF.

[2] Senneville, Z. Albalawi, S. A. van Asten, Z. G. Abbas, G. Allison, J. Aragón-Sánchez, J. M. Embil, L. A. Lavery, M. Alhasan, O. Oz, I. Uçkay, V. Urbančič-Rovan, Z.-R. Xu, and E. J. Peters, "Iwgdf/idsa guidelines on the diagnosis and treatment of diabetes-related foot infections (iwgdf/idsa 2023)," *Clinical Infectious Diseases*, 2023. [Online]. Available: https://doi.org/10.1093/cid/ciad527

[3] F. S. Southwick, "Infectious diseases: a clinical short course," *(No Title)*, 2020.

[4] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[6] F. Veredas, H. Mesa, and L. Morente, "Binary tissue classification on wound images with neural networks and bayesian classifiers," *IEEE transactions on medical imaging*, vol. 29, no. 2, pp. 410–427, 2009.

[7] H. Wannous, Y. Lucas, and S. Treuillet, "Enhanced assessment of the wound-healing process by accurate multiview tissue classification," *IEEE transactions on Medical Imaging*, vol. 30, no. 2, pp. 315–326, 2010.

[8] L. Wang, P. C. Pedersen, E. Agu, D. M. Strong, and B. Tulu, "Area determination of diabetic foot ulcer images using a cascaded two-stage svm-based classification," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2098–2109, 2016.

[9] M. H. Yap, M. Goyal, F. M. Osman, R. Martí, E. Denton, A. Juette, and R. Zwiggelaar, "Breast ultrasound lesions recognition: end-to-end deep learning approaches," *Journal of medical imaging*, vol. 6, no. 1, pp. 011 007–011 007, 2019.

[10] E. Ahmad, M. Goyal, J. S. McPhee, H. Degens, and M. H. Yap, "Semantic segmentation of human thigh quadriceps muscle in magnetic resonance images," *arXiv preprint arXiv:1801.00415*, 2018.

[11] M. Goyal, N. D. Reeves, A. K. Davison, S. Rajbhandari, J. Spragg, and M. H. Yap, "Dfunet: Convolutional neural networks for diabetic foot ulcer classification," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, pp. 728–739, 2018.

[12] M. Goyal, M. H. Yap, N. D. Reeves, S. Rajbhandari, and J. Spragg, "Fully convolutional networks for diabetic foot ulcer segmentation," in *2017 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2017, pp. 618–623.

[13] M. Goyal, N. D. Reeves, S. Rajbhandari, and M. H. Yap, "Robust methods for real-time diabetic foot ulcer detection and localization on mobile devices," *IEEE journal of biomedical and health informatics*, vol. 23, no. 4, pp. 1730–1741, 2018.

[14] C. Wang, X. Yan, M. Smith, K. Kochhar, M. Rubin, S. M. Warren, J. Wrobel, and H. Lee, "A unified framework for automatic wound segmentation and analysis with deep convolutional neural networks," in *2015 37th annual international conference of the ieee engineering in medicine and biology society (EMBC)*. IEEE, 2015, pp. 2415–2418.

[15] M. Goyal, N. D. Reeves, S. Rajbhandari, N. Ahmad, C. Wang, and M. H. Yap, "Recognition of ischaemia and infection in diabetic foot ulcers: Dataset and techniques," *Computers in biology and medicine*, vol. 117, p. 103616, 2020.

[16] K. Cabello-Solorzano, I. Ortigosa de Araujo, M. Peña, L. Correia, and A. J. Tallón-Ballesteros, "The impact of data normalization on the accuracy of machine learning algorithms: A comparative analysis," in *International conference on soft computing models in industrial and environmental applications*. Springer, 2023, pp. 344–353.

[17] M. Dziuba, I. Jarsky, V. Efimova, and A. Filchenkov, "Image vectorization: a review," *Journal of Mathematical Sciences*, vol. 285, no. 2, pp. 155–168, 2024.

[18] V. Schmidt, B. Courty, and A. Saboni, "Codecarbon," *URL: https://github. com/mlco2/codecarbon. accessed*, pp. 11–21, 2023.

[19] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 09, 2020, pp. 13 693–13 696.

[20] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, "Towards the systematic reporting of the energy and carbon footprints of machine learning," *Journal of Machine Learning Research*, vol. 21, no. 248, pp. 1–43, 2020.

[21] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, "Quantifying the carbon emissions of machine learning," *arXiv preprint arXiv:1910.09700*, 2019.

[22] W. O. de Araujo and C. J. Coelho, "Análise de componentes principais (pca)," *University Center of Anápolis, Annapolis*, 2009.

[23] F. E. de Melo Borges, O. F. Mota, D. D. Ferreira, and B. H. G. Barbosa, "One-class classifier based on principal curves," *Neural Computing and Applications*, vol. 35, no. 26, pp. 19 015–19 024, 2023.

[24] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[25] A. I. A. Osman, A. N. Ahmed, M. F. Chow, Y. F. Huang, and A. El-Shafie, "Extreme gradient boosting (xgboost) model to predict the groundwater levels in selangor malaysia," *Ain Shams Engineering Journal*, vol. 12, no. 2, pp. 1545–1556, 2021.

[26] S. Haykin, *Redes neurais: princípios e prática*. Bookman Editora, 2001.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2. Montreal, Canada, 1995, pp. 1137–1145.

[29] F. E. d. M. Borges, "OCPC-PY: One Class Classifier based on Principal Curves in Python," https://pypi.org/project/ocpc-py/, 2023, disponível em: https://pypi.org/project/ocpc-py/.

[30] P. Fabian, "Scikit-learn: Machine learning in python," *Journal of machine learning research 12*, p. 2825, 2011.