



UNIVERSIDADE FEDERAL DO CEARÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

PEDRO SOUSA DE ARAGÃO NETO

MÉTODOS DE APRENDIZADO DE MÁQUINA APLICADOS AO
MONITORAMENTO AMBIENTAL

SOBRAL

2026

PEDRO SOUSA DE ARAGÃO NETO

MÉTODOS DE APRENDIZADO DE MÁQUINA APLICADOS AO MONITORAMENTO
AMBIENTAL

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Engenharia de
Computação da Universidade Federal do Ceará,
como requisito parcial à obtenção do grau de
em Engenharia de Computação.

Orientador: Prof. Dr. Thiago Iachiley
Araujo de Souza

SOBRAL

2026

PEDRO SOUSA DE ARAGÃO NETO

MÉTODOS DE APRENDIZADO DE MÁQUINA APLICADOS AO MONITORAMENTO
AMBIENTAL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de em Engenharia de Computação.

Aprovada em: 27/01/2026

BANCA EXAMINADORA

Prof. Dr. Thiago Iachiley Araujo de
Souza (Orientador)
Curso Engenharia de Computação - Campus da UFC
em Sobral

Prof. Dr. Iális Cavalcante de Paula Júnior
Curso Engenharia de Computação - Campus da UFC
em Sobral

Prof. Me. Mateus Alcântara De Castro Borges
Secretaria Estadual De Educação Do Ceará – Seduc

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

A672m Aragão Neto, Pedro Sousa de.
Métodos de aprendizagem de máquina aplicados ao monitoramento ambiental / Pedro Sousa de Aragão Neto. – 2026.
50 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Sobral, Curso de Engenharia da Computação, Sobral, 2026.
Orientação: Prof. Dr. Thiago Iachiley Araujo de Souza.

1. Aprendizagem de máquina. 2. Redes neurais. 3. Monitoramento ambiental. 4. Classificação. 5. Regressão. I. Título.

CDD 621.39

Dedico este trabalho a todas as pessoas que caminharam conosco ao longo desta trajetória, acreditaram em nossos sonhos e nos sustentaram nos momentos mais difíceis, oferecendo apoio, compreensão, paciência e amor. Sem vocês, esta conquista não seria possível.

AGRADECIMENTOS

Agradeço a Deus, pois sabemos que ele está presente em nossas vidas e que sem ele nada seria possível.

Agradeço a minha esposa Nara Ney Almeida de Lima que estava comigo em todos os momentos, me apoiando, me incentivando e por ela acreditar na conquista desse sonho.

Agradeço minha mãe Teresa de Melo Soares, meu pai Reginaldo Ferreira Aragão e meu irmão Ronielly Alves Aragão, pelo apoio e por acreditarem nessa conquista.

Agradecemos aos amigos que durante esse tempo estiveram ao meu lado nos momentos felizes e também nos difíceis.

Agradecemos aos professores pela dedicação e pelo incentivo para que continuássemos até o fim.

Ao meu orientador professor Dr. Thiago Iachhiley Araujo de Souza pela paciência e dedicação durante o desenvolvimento desse trabalho, meus sinceros agradecimentos.

“Para realizar grandes conquistas, devemos não apenas agir, mas também sonhar, não apenas planejar, mas também acreditar.”

(Anatole France)

RESUMO

Nos últimos anos, no mundo vem ocorrendo transformações gigantescas no meio ambiente e onde se têm uma ocorrência de grandes inovações tecnológicas sempre surgindo, então surge a necessidade de como a tecnologia pode ser benéfica para o meio ambiente. As Redes Neurais surgiram como ferramentas poderosas para a classificação de dados complexos, oferecendo soluções para a identificação de padrões e anomalias. Este artigo explora a aplicação de três redes neurais – Redes Neurais Artificiais (ANN), Perceptron Multicamadas (MLP) e Redes de Função de Base Radial (RBF) – no contexto da classificação de dados ambientais urbanos. Apresentamos uma análise comparativa de seu desempenho utilizando conjuntos de dados reais, destacando suas forças e limitações na categorização precisa de parâmetros ambientais. Nossos resultados demonstram a eficácia desses modelos de redes neurais em contribuir no monitoramento do meio ambiente, fornecendo insights valiosos.

Palavras-chave: Aprendizado de máquina. Inteligência artificial. Redes neurais artificiais. Monitoramento ambiental. Classificação e previsão de dados.

ABSTRACT

In recent years, the world has been undergoing enormous environmental transformations and where there is a constant occurrence of major technological innovations, the need arises to understand how technology can be beneficial to the environment. Neural Networks have emerged as powerful tools for classifying complex data, offering solutions for identifying patterns and anomalies. This article explores the application of three neural networks – Artificial Neural Networks (ANN), Multilayer Perceptron (MLP) and Radial Basis Function Networks (RBF) – in the context of urban environmental data classification. We present a comparative analysis of their performance using real datasets, highlighting their strengths and limitations in accurately categorizing environmental parameters. Our results demonstrate the effectiveness of these neural network models in contributing to environmental monitoring, providing valuable insights.

Keywords: Machine learning. Artificial intelligence. Artificial neural networks. Environmental monitoring. Data classification and prediction.

LISTA DE TABELAS

Tabela 1 – Relatório de Classificação RBF	41
Tabela 2 – Relatório de Classificação MLP	41
Tabela 3 – Relatório de Classificação ANN	42
Tabela 4 – Métricas de Desempenho para Previsão	44

LISTA DE ABREVIATURAS E SIGLAS

ANN	Rede Neural Artificial
IA	Inteligência Artificial
IOT	Internet das Coisas
ML	Machine Learning
MLP	Perceptrons Multicamadas
RBF	Redes de Função de Base Radial
RNAs	Redes Neurais Artificiais

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivo Geral	14
<i>1.1.1</i>	<i>Objetivos Específicos</i>	<i>14</i>
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Histórico e Evolução das Redes Artificiais	16
2.2	Fundamentos Matemáticos das Redes Neurais	18
<i>2.2.1</i>	<i>Modelo Matemático do Neurônio Artificial</i>	<i>18</i>
<i>2.2.2</i>	<i>Funções de Ativação</i>	<i>19</i>
<i>2.2.3</i>	<i>Algoritmo de Backpropagation</i>	<i>20</i>
2.3	Redes Neurais Artificiais (ANN)	20
<i>2.3.1</i>	<i>Arquitetura e Componentes das ANNs</i>	<i>20</i>
<i>2.3.2</i>	<i>Processo de Treinamento das ANNs</i>	<i>21</i>
2.4	Multilayer Perceptrons (MLP)	23
<i>2.4.1</i>	<i>Teorema da Aproximação Universal</i>	<i>23</i>
<i>2.4.2</i>	<i>Arquitetura Detalhada das MLPs</i>	<i>23</i>
<i>2.4.3</i>	<i>Processo de Treinamento Detalhado</i>	<i>24</i>
<i>2.4.4</i>	<i>Desafios no Treinamento de MLPs</i>	<i>25</i>
<i>2.4.5</i>	<i>Aplicações Específicas das MLPs em Monitoramento Ambiental</i>	<i>25</i>
2.5	Redes de Função de Base Radial (RBF)	25
<i>2.5.1</i>	<i>Fundamentos Teóricos das Redes RBF</i>	<i>26</i>
<i>2.5.2</i>	<i>Arquitetura das Redes RBF</i>	<i>26</i>
<i>2.5.3</i>	<i>Processo de Treinamento das Redes RBF</i>	<i>27</i>
<i>2.5.4</i>	<i>Vantagens e Desvantagens das Redes RBF</i>	<i>27</i>
<i>2.5.5</i>	<i>Aplicações das Redes RBF em Monitoramento Ambiental</i>	<i>28</i>
2.6	Técnicas de Validação e Avaliação de Modelos	28
<i>2.6.1</i>	<i>Conjunto de Treinamento, Validação e Teste</i>	<i>28</i>
<i>2.6.2</i>	<i>Métricas de Avaliação</i>	<i>29</i>
3	METODOLOGIA	30
3.1	Coleta e Caracterização dos Dados	30
3.2	Pré-Processamento de Dados	32

3.2.1	<i>Limpeza de Dados</i>	32
3.2.2	<i>Normalização/Padronização</i>	33
3.2.3	<i>Codificação de Variáveis Categóricas</i>	33
3.2.4	<i>Engenharia de Features (Feature Engineering)</i>	33
3.3	Divisão dos Dados	34
3.4	Implementação dos Modelos de Redes Neurais	35
3.4.1	<i>Configuração das Redes Neurais</i>	35
3.5	Treinamento e Avaliação dos Modelos	38
3.5.1	<i>Métricas de Classificação</i>	38
3.5.2	<i>Métricas de Previsão (Regressão)</i>	38
3.6	Ferramentas e Ambiente de Desenvolvimento	38
4	RESULTADOS E DISCUSSÃO	41
4.1	Resultados de Classificação	41
4.2	Resultados da Regressão	44
5	CONSIDERAÇÕES FINAIS	47
	REFERÊNCIAS	48

1 INTRODUÇÃO

O monitoramento ambiental é uma área essencial para a sustentabilidade do planeta, pois envolve a coleta, o processamento e a análise de dados sobre o estado do meio ambiente (SILVA, 2020). Tradicionalmente, essa atividade demandava grande quantidade de recursos e tempo, uma vez que dependia de métodos manuais e análises laboratoriais. Contudo, com o avanço tecnológico, especialmente no campo da Inteligência Artificial (Inteligência Artificial (IA)) e do Aprendizado de Máquina (Machine Learning – Machine Learning (ML)), novas abordagens vêm sendo aplicadas para otimizar e aprimorar esses processos (SOUZA; OLIVEIRA, 2021).

A aplicação de métodos de aprendizado de máquina no monitoramento ambiental permite processar grandes volumes de dados provenientes de sensores, imagens de satélite e outras fontes, identificando padrões complexos e prevendo eventos futuros com maior precisão e eficiência (PEREIRA; COSTA, 2022). Essa integração tecnológica oferece um suporte estratégico para a formulação de políticas públicas, o gerenciamento de recursos naturais e a mitigação de impactos ambientais (SANTOS; LIMA, 2021). Em um cenário global marcado por mudanças climáticas, perda de biodiversidade e escassez hídrica, a capacidade de monitorar e responder rapidamente a alterações ambientais tornou-se indispensável.

O desenvolvimento de sensores de baixo custo, plataformas de sensoriamento remoto (como satélites e drones) e a conectividade proporcionada pela Internet das Coisas (Internet das Coisas (IOT)) resultaram em uma explosão na quantidade e diversidade de dados ambientais disponíveis (RODRIGUES; ALMEIDA, 2022). Essa abundância, no entanto, trouxe o desafio de transformar dados brutos em conhecimento útil. Nesse contexto, a IA, com sua capacidade de analisar grandes volumes de informação, identificar padrões e realizar previsões, surge como uma ferramenta fundamental para compreender fenômenos ambientais complexos (ALMEIDA; FERREIRA, 2022).

Redes neurais artificiais (Redes Neurais Artificiais (RNAs)) se destacam nesse cenário por sua habilidade em modelar relações não lineares e aprender diretamente a partir dos dados (GOODFELLOW *et al.*, 2016). Inspiradas no funcionamento do cérebro humano, essas redes vêm sendo amplamente utilizadas em aplicações que envolvem reconhecimento de padrões, previsão de séries temporais e classificação de dados. No monitoramento ambiental, as RNAs têm se mostrado eficazes na previsão da qualidade do ar, detecção de eventos extremos e análise de poluentes (FERNANDES *et al.*, 2020).

Entre as principais arquiteturas de RNAs, destacam-se as Redes Neurais Artificiais (Rede Neural Artificial (ANN)), os Perceptrons Multicamadas (Perceptrons Multicamadas (MLP)) e as Redes de Função de Base Radial (Redes de Função de Base Radial (RBF)). As ANNs representam a base conceitual dessas arquiteturas e têm sido aplicadas com sucesso em diferentes áreas. As MLPs, com suas camadas ocultas e funções de ativação não lineares, são capazes de aprender representações hierárquicas e identificar padrões complexos nos dados. Já as RBFs, que utilizam funções gaussianas centradas em pontos específicos do espaço de entrada, são especialmente eficazes em problemas de interpolação e classificação (BISHOP, 2006).

O presente trabalho tem como objetivo explorar a aplicação dessas três arquiteturas de redes neurais no contexto do monitoramento ambiental, com foco em tarefas de classificação e previsão. Busca-se demonstrar como essas técnicas podem contribuir para uma compreensão mais profunda dos fenômenos ambientais e para uma gestão mais eficaz dos recursos naturais. Além disso, pretende-se realizar uma análise comparativa do desempenho de cada modelo, destacando suas vantagens, limitações e potenciais combinações para futuras pesquisas.

A relevância deste estudo reside na necessidade crescente de soluções inovadoras que unam sustentabilidade e tecnologia. Ao aplicar métodos de aprendizado de máquina em dados ambientais, é possível obter insights que favoreçam decisões mais assertivas e estratégias de preservação mais eficazes. Assim, este trabalho contribui para o avanço do uso de técnicas de IA no setor ambiental, fortalecendo a integração entre ciência, tecnologia e sustentabilidade.

1.1 Objetivo Geral

Investigar e analisar a aplicação de diferentes métodos de Aprendizagem de Máquina, especificamente Redes Neurais Artificiais (ANN), Multilayer Perceptrons (MLP) e Redes de Função de Base Radial (RBF), em tarefas de classificação e previsão no contexto do monitoramento de variáveis ambientais, nas quais o sistema realiza o acompanhamento de parâmetros como temperatura e concentração de gases.

1.1.1 Objetivos Específicos

1. Analisar a fundamentação teórica das arquiteturas ANN, MLP e RBF, compreendendo seus princípios e algoritmos de treinamento.
2. Implementar e configurar os modelos para tarefas de classificação e previsão utilizando

dados de monitoramento ambiental.

3. Avaliar o desempenho das arquiteturas utilizando métricas estatísticas apropriadas para cada tarefa.
4. Comparar os resultados obtidos para identificar as vantagens e limitações de cada modelo em cenários ambientais.
5. Discutir as implicações práticas dos resultados e propor melhorias para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção aborda os conceitos fundamentais das redes neurais artificiais (RNAs), com foco nas arquiteturas ANN, MLP e RBF, que são a base para as aplicações exploradas neste trabalho. Compreender o funcionamento dessas redes é crucial para apreciar sua utilidade no monitoramento ambiental. A fundamentação teórica apresentada aqui fornece a base conceitual necessária para entender como essas tecnologias podem ser aplicadas efetivamente em problemas de monitoramento ambiental, desde a coleta e processamento de dados até a geração de *insights* acionáveis para a gestão sustentável dos recursos naturais (HAYKIN, 2009).

2.1 Histórico e Evolução das Redes Artificiais

O avanço das Redes Neurais Artificiais (RNAs) teve início na década de 1940, com a proposta do primeiro modelo matemático de um neurônio artificial, denominado neurônio de McCulloch-Pitts, por McCulloch e Pitts (1943). Esse modelo serviu como base conceitual para o desenvolvimento de arquiteturas mais complexas, propondo que o cérebro humano poderia ser representado por um sistema de unidades interligadas que processam sinais elétricos e químicos (MCCULLOCH; PITTS, 1943).

A evolução das redes neurais pode ser dividida em várias fases distintas. A primeira fase, nas décadas de 1940 e 1950, foi marcada pelo desenvolvimento do perceptron por Frank Rosenblatt (1958), que demonstrou a capacidade de uma rede neural simples de aprender a classificar padrões lineares. No entanto, as limitações do perceptron, especialmente sua incapacidade de resolver problemas não-lineares como a função XOR, levaram ao que ficou conhecido como o "inverno da IA" nas décadas de 1960 e 1970 (ROSENBLATT, 1958).

O renascimento das redes neurais ocorreu na década de 1980, impulsionado principalmente pelo desenvolvimento do algoritmo de *backpropagation* por Rumelhart, Hinton e Williams (1986). Este algoritmo permitiu o treinamento eficiente de redes neurais multicamadas, superando as limitações dos perceptrons simples e abrindo caminho para aplicações mais complexas (RUMELHART *et al.*, 1986).

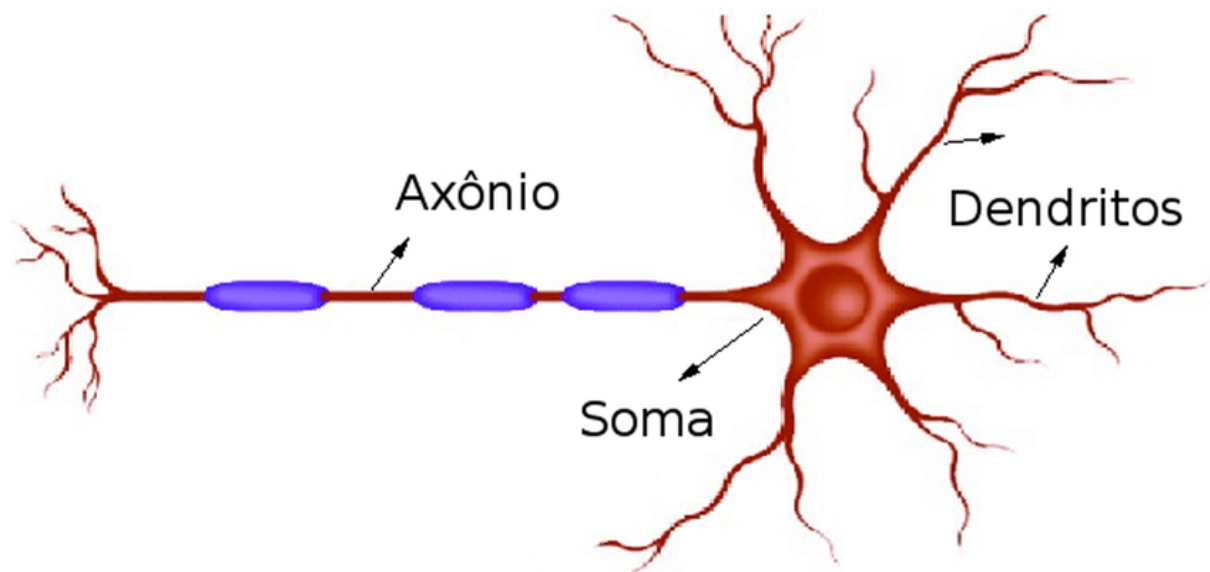
A década de 1990 viu o desenvolvimento de arquiteturas especializadas, incluindo as redes de função de base radial (RBF), que ofereciam vantagens específicas em problemas de aproximação de funções e classificação. Simultaneamente, avanços em teoria de aprendizado estatístico, como as máquinas de vetores de suporte (SVM), forneceram alternativas competitivas

às redes neurais em muitas aplicações (BROOMHEAD; LOWE, 1988).

Simultaneamente, pesquisas como a de Borges *et al.* (2015) ampliaram a compreensão do comportamento dinâmico dos neurônios e das redes biológicas, enfatizando a importância da plasticidade sináptica, referente a habilidade das conexões entre os neurônios de se fortalecerem ou enfraquecerem de acordo com a atividade neural. Esse aspecto, identificado no modelo *Hodgkin-Huxley*, fornece uma base matemática para a comparação entre neurônios biológicos e artificiais, estabelecendo uma conexão entre neurociência e aprendizado de máquina. Além disso, o estudo mostrou que a sincronização de disparos neuronais pode ser simulada por meio de modelos computacionais, o que afeta a forma como as redes neurais artificiais processam e aprendem padrões.

Um modelo neuronal tenta replicar, de forma simplificada, o funcionamento de um neurônio biológico, que é a célula encarregada de transmitir impulsos nervosos e facilitar a comunicação entre diversas áreas do sistema nervoso. O cérebro humano é composto por bilhões dessas células interligadas, que criam uma rede altamente complexa e dinâmica.

Figura 1 – Componentes básicos de um neurônio.



Fonte: (BORGES *et al.*, 2015)

Cada neurônio é constituído por três componentes principais: o soma, que contém o núcleo e é responsável pelas funções metabólicas da célula; os dendritos, que atuam como receptores de sinais de outros neurônios; e o axônio, responsável por conduzir impulsos elétricos para outras células. Essa estrutura possibilita o processamento simultâneo de informações, o que significa que diversos estímulos podem ser recebidos e processados ao mesmo tempo,

gerando respostas ágeis e adaptativas. A representação esquemática desses componentes pode ser observada na Figura 7, que ilustra a organização básica de um neurônio biológico e suas principais partes.

A compreensão dessa dinâmica inspirou o desenvolvimento das redes neurais artificiais, nas quais os neurônios biológicos são representados por unidades matemáticas capazes de receber, processar e transmitir informações. Assim como ocorre no cérebro, cada unidade artificial interage com outras por meio de “conexões”, os pesos sinápticos, que determinam a força e a influência de um sinal. Essa analogia entre biologia e computação possibilitou a criação de modelos capazes de aprender a partir de dados, generalizar padrões e oferecer soluções automatizadas para problemas complexos, como a análise e o monitoramento ambiental (BORGES *et al.*, 2015; HAYKIN, 2009).

O século XXI trouxe uma revolução no campo das redes neurais com o advento do *deep learning*. O aumento da capacidade computacional, a disponibilidade de grandes conjuntos de dados e melhorias algorítmicas levaram ao desenvolvimento de redes neurais profundas capazes de resolver problemas complexos em visão computacional, processamento de linguagem natural e muitas outras áreas (LECUN *et al.*, 2015).

Por fim, como destaca Paixão (2024), as RNAs se consolidaram como modelos capazes de processar informações de forma distribuída e paralela, aprendendo padrões não lineares e apresentando grande capacidade de generalização. Elas representam uma das mais importantes ferramentas do aprendizado de máquina, aplicáveis a múltiplas áreas, desde da engenharia elétrica à sustentabilidade ambiental, servindo como base para a automação inteligente e o desenvolvimento de tecnologias de monitoramento mais precisas e eficientes .

2.2 Fundamentos Matemáticos das Redes Neurais

2.2.1 Modelo Matemático do Neurônio Artificial

O neurônio artificial é a unidade básica de processamento em uma rede neural. Matematicamente, um neurônio pode ser descrito pela seguinte equação:

$$y = f \left(\sum_{i=1}^n w_i x_i + b \right)$$

Onde:

- x_i são as entradas do neurônio;

- w_i são os pesos sinápticos correspondentes;
- b é o bias (viés);
- f é a função de ativação;
- y é a saída do neurônio;

Esta formulação simples encapsula o processamento fundamental que ocorre em cada neurônio: uma combinação linear das entradas seguida por uma transformação não-linear através da função de ativação (BISHOP, 2006).

2.2.2 Funções de Ativação

As funções de ativação desempenham um papel central no funcionamento das redes neurais artificiais, pois são responsáveis por introduzir não linearidade no modelo, permitindo que a rede aprenda e represente relações complexas presentes nos dados. A escolha adequada da função de ativação influencia diretamente a capacidade de aprendizado, a velocidade de convergência e o desempenho final da rede.

Entre as funções mais tradicionais está a função sigmoide, definida por $\sigma(x) = \frac{1}{1 + e^{-x}}$, essa função mapeia qualquer valor real para o intervalo $(0, 1)$, o que a torna especialmente útil em problemas de classificação binária, nos quais a saída pode ser interpretada como uma probabilidade. Entretanto, a sigmoide apresenta limitações importantes, como o problema do gradiente evanescente, em que os gradientes assumem valores muito pequenos durante o processo de backpropagation, dificultando o treinamento de redes mais profundas (GLOROT; BENGIO, 2010).

Outra função amplamente utilizada é a tangente hiperbólica, definida por $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, embora seja semelhante à sigmoide, a função tangente hiperbólica mapeia os valores de entrada para o intervalo $(-1, 1)$, o que pode resultar em uma convergência ligeiramente melhor devido ao fato de suas saídas serem centradas em zero. Ainda assim, essa função também sofre com o problema do gradiente evanescente, especialmente em redes profundas.

Com o avanço das arquiteturas profundas, a função ReLU (Rectified Linear Unit) tornou-se uma das mais utilizadas. Ela é definida por $\text{ReLU}(x) = \max(0, x)$, a simplicidade da ReLU, aliada à sua eficiência computacional, contribui para mitigar o problema do gradiente evanescente e acelera o treinamento de redes profundas. Apesar de suas vantagens, a ReLU pode apresentar limitações, como neurônios que deixam de ativar durante o treinamento. Para contornar esses problemas, variações como a Leaky ReLU e a ELU foram propostas, mantendo

as vantagens da função original e reduzindo seus efeitos adversos (NAIR; HINTON, 2010).

2.2.3 Algoritmo de Backpropagation

O algoritmo de backpropagation é o método padrão para treinar redes neurais multicamadas. O algoritmo funciona em duas fases:

1. **Forward Pass:** Os dados de entrada são propagados através da rede para calcular a saída.
2. **Backward Pass:** O erro é calculado e propagado de volta através da rede para atualizar os pesos.

Matematicamente, a atualização dos pesos é dada por $w_{ij}^{(l)} = w_{ij}^{(l)} - \eta \frac{\partial E}{\partial w_{ij}^{(l)}}$, onde η é a taxa de aprendizado e $\frac{\partial E}{\partial w_{ij}^{(l)}}$ é o gradiente da função de erro em relação ao peso $w_{ij}^{(l)}$ na camada l (GOODFELLOW *et al.*, 2016).

2.3 Redes Neurais Artificiais (ANN)

As Redes Neurais Artificiais (ANNs) representam uma das abordagens mais fundamentais e amplamente utilizadas em aprendizado de máquina. Inspiradas na estrutura e funcionamento do sistema nervoso biológico, as ANNs são compostas por unidades de processamento interconectadas (neurônios artificiais) organizadas em camadas hierárquicas. Esta arquitetura permite que as redes aprendam representações complexas dos dados através de um processo de treinamento supervisionado ou não supervisionado (ANDERSON; ROSENFELD, 1988).

2.3.1 Arquitetura e Componentes das ANNs

A arquitetura básica de uma rede neural artificial é composta por três tipos principais de camadas, cada uma desempenhando um papel específico no processamento da informação. A camada de entrada é responsável por receber os dados brutos provenientes do ambiente externo, sendo que cada neurônio dessa camada corresponde a uma característica dos dados de entrada. Em aplicações de monitoramento da qualidade da água, por exemplo, esses neurônios podem representar parâmetros como pH, temperatura, oxigênio dissolvido e turbidez. Dessa forma, o número de neurônios na camada de entrada é diretamente determinado pela dimensionalidade do espaço de características do problema em análise (MITCHELL, 1997).

As camadas ocultas concentram a maior parte do processamento e do aprendizado

da rede. É nelas que os dados de entrada são transformados em representações internas mais abstratas, permitindo a extração de características relevantes para a resolução do problema. Tanto o número de camadas ocultas quanto a quantidade de neurônios em cada uma constituem hiperparâmetros fundamentais, pois influenciam diretamente a capacidade de aprendizado e a habilidade de generalização da rede. A profundidade, entendida como o número de camadas ocultas, está relacionada à complexidade das funções que a rede pode aproximar. Redes mais profundas são capazes de representar funções mais complexas, porém tendem a ser mais difíceis de treinar e mais suscetíveis ao overfitting. Já a largura da rede, definida pelo número de neurônios por camada, afeta a capacidade de representação em cada nível de abstração (CYBENKO, 1989).

Por fim, a camada de saída é responsável por produzir o resultado final da rede neural, e sua configuração depende diretamente do tipo de problema abordado. Em problemas de regressão, é comum a utilização de um único neurônio com função de ativação linear. Para tarefas de classificação binária, emprega-se geralmente um neurônio com função de ativação sigmoide, enquanto em problemas de classificação multiclasse são utilizados múltiplos neurônios, um para cada classe, associados a uma função de ativação softmax.

2.3.2 *Processo de Treinamento das ANNs*

O treinamento de uma ANN é um processo iterativo cujo objetivo é ajustar os pesos e vieses da rede de modo a minimizar uma função de custo. Esse processo é composto por diversas etapas interligadas, que vão desde a inicialização dos parâmetros até a aplicação de técnicas avançadas de otimização e regularização, garantindo aprendizado eficiente e boa capacidade de generalização.

A etapa inicial consiste na inicialização dos pesos, considerada crucial para o sucesso do treinamento. Entre os métodos mais utilizados estão a inicialização aleatória, na qual os pesos recebem valores pequenos e aleatórios, a inicialização Xavier ou Glorot, projetada para manter a variância das ativações aproximadamente constante ao longo das camadas, e a inicialização He, especialmente indicada para redes que utilizam funções de ativação do tipo ReLU.

Em seguida, ocorre a propagação direta dos dados pela rede, conhecida como forward propagation. Nessa fase, as entradas são processadas camada a camada até a produção da saída final. Para cada camada l , a ativação é calculada por meio da combinação linear entre os pesos, as ativações da camada anterior e os vieses, seguida da aplicação da função de ativação, conforme a expressão $a^{(l)} = f^{(l)}(\mathbf{W}^{(l)}a^{(l-1)} + b^{(l)})$, onde $W^{(l)}$ representa a matriz de pesos, $b^{(l)}$ o vetor de

vieses e $f^{(l)}$ a função de ativação da camada l .

Após a obtenção da saída da rede, calcula-se o erro comparando os valores previstos com os valores desejados. Essa comparação é realizada por meio de uma função de custo adequada ao tipo de problema. Em tarefas de regressão, são frequentemente utilizados o erro quadrático médio (MSE) e o erro absoluto médio (MAE), enquanto em problemas de classificação a entropia cruzada é a escolha mais comum.

O ajuste dos parâmetros é realizado por meio da retropropagação do erro (backward propagation). Nessa etapa, o algoritmo de backpropagation calcula os gradientes da função de custo em relação a todos os pesos e vieses da rede, permitindo atualizar os parâmetros na direção que minimiza o erro global. A atualização básica dos parâmetros é realizada utilizando a taxa de aprendizado η .

Para melhorar a capacidade de generalização e reduzir o sobreajuste, técnicas de regularização são incorporadas à função de custo. A regularização L1 adiciona um termo de penalidade da forma $\lambda \sum_i |w_i|$, que incentiva soluções esparsas, enquanto a regularização L2 utiliza $\lambda \sum_i w_i^2$, penalizando pesos de grande magnitude e contribuindo para maior estabilidade do modelo. Outra técnica amplamente utilizada é o dropout, no qual, durante o treinamento, uma fração aleatória dos neurônios é temporariamente desativada, reduzindo a dependência excessiva de unidades específicas e melhorando a robustez do modelo (SRIVASTAVA *et al.*, 2014). O early stopping atua de forma complementar, interrompendo o treinamento quando o desempenho no conjunto de validação começa a se degradar. Já a batch normalization normaliza as ativações de cada camada em mini-batches, acelerando o treinamento e aumentando a estabilidade do processo (IOFFE; SZEGEDY, 2015).

O processo de otimização pode ser aprimorado por meio do uso de otimizadores mais sofisticados. O método do momentum introduz uma memória nas atualizações dos pesos, sendo definido pelas equações: $v_t = \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta)$ e $\theta = \theta - v_t$, o que contribui para acelerar a convergência e superar mínimos locais. O AdaGrad adapta a taxa de aprendizado individualmente para cada parâmetro com base no histórico dos gradientes, conforme: $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$. Já o Adam (Adaptive Moment Estimation) combina as vantagens do momentum e do AdaGrad, mantendo médias móveis exponenciais dos gradientes e de seus quadrados, sendo descrito por:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

Devido à sua capacidade de modelar relações não lineares complexas e lidar com dados ruidosos e incompletos, as redes neurais artificiais têm sido amplamente aplicadas no mo-

nitoramento ambiental. Na previsão da qualidade do ar, essas redes podem estimar concentrações futuras de poluentes atmosféricos a partir de dados meteorológicos, de tráfego e de emissões industriais, explorando a interação entre múltiplas variáveis (GARDNER; DORLING, 1998). No monitoramento da qualidade da água, ANNs são empregadas para prever parâmetros como demanda bioquímica de oxigênio, sólidos suspensos totais e concentrações de nutrientes com base em medições mais facilmente obtidas (PALANI *et al.*, 2008). Além disso, no contexto do sensoriamento remoto, essas redes são utilizadas para detectar e classificar mudanças no uso e cobertura do solo ao longo do tempo, fornecendo subsídios relevantes para a gestão territorial e a conservação ambiental (PIJANOWSKI *et al.*, 2002).

2.4 Multilayer Perceptrons (MLP)

Os Multilayer Perceptrons (MLPs) representam uma extensão natural das redes neurais simples, incorporando uma ou mais camadas ocultas entre a entrada e a saída. Esta arquitetura multicamadas confere às MLPs a capacidade de aproximar funções não-lineares arbitrariamente complexas, tornando-as uma das arquiteturas mais versáteis e amplamente utilizadas em aprendizado de máquina (HORNIK *et al.*, 1989).

2.4.1 Teorema da Aproximação Universal

Um dos resultados teóricos mais importantes relacionados às MLPs é o Teorema da Aproximação Universal, que estabelece que uma MLP com uma única camada oculta contendo um número suficiente de neurônios pode aproximar qualquer função contínua em um domínio compacto com precisão arbitrária. Formalmente, para qualquer função contínua $f : [a, b]^n \rightarrow \mathbb{R}$ e qualquer $\varepsilon > 0$, existe uma MLP F , tal que $|f(x) - F(x)| < \varepsilon$, para todo x no domínio (CYBENKO, 1989). Este teorema fornece a base teórica para a capacidade das MLPs de resolver uma ampla variedade de problemas, embora não forneça orientação sobre como determinar o número necessário de neurônios ocultos na prática.

2.4.2 Arquitetura Detalhada das MLPs

As MLPs possuem características estruturais que definem seu funcionamento e seu potencial de modelagem. Uma dessas características é a conectividade feedforward, na qual as conexões entre os neurônios seguem uma única direção, partindo da camada de entrada em

direção à camada de saída, sem a formação de ciclos. Essa organização torna o processo de treinamento mais simples e garante que a computação realizada pela rede seja determinística e eficiente.

Outro aspecto fundamental das MLPs é o uso de camadas densamente conectadas. Em uma arquitetura tradicional, cada neurônio de uma camada está ligado a todos os neurônios da camada subsequente, o que possibilita à rede capturar interações complexas entre as características de entrada. Entretanto, essa conectividade total também implica em um número elevado de parâmetros, principalmente à medida que o número de camadas e de neurônios aumenta, o que pode elevar o custo computacional e o risco de sobreajuste.

Além disso, as MLPs dependem fortemente do uso de funções de ativação não lineares aplicadas em cada neurônio. Essas funções são responsáveis por introduzir não linearidade no modelo, permitindo que a rede represente relações complexas entre entrada e saída. Sem elas, mesmo uma MLP com múltiplas camadas se comportaria como uma simples transformação linear, limitando significativamente sua capacidade de aprendizado.

2.4.3 *Processo de Treinamento Detalhado*

O processo de treinamento de uma rede Multilayer Perceptron é baseado no algoritmo de backpropagation, que fornece um procedimento sistemático para o ajuste dos pesos e vieses da rede. Esse algoritmo pode ser descrito de forma matemática a partir de três etapas principais. Inicialmente, ocorre o forward pass, no qual os dados de entrada são propagados camada a camada, desde a entrada até a saída da rede. Para cada camada $l = 1, 2, \dots, L$, calcula-se o potencial interno por meio da combinação linear entre os pesos, as ativações da camada anterior e o viés, dada por $z^{(l)} = \mathbf{W}^{(l)}a^{(l-1)} + b^{(l)}$. Em seguida, aplica-se a função de ativação correspondente, resultando em $a^{(l)} = f^{(l)}(z^{(l)})$.

Após a propagação direta, inicia-se o backward pass, que consiste na propagação do erro da camada de saída em direção às camadas anteriores. Esse processo começa pela camada final, onde o termo de erro é calculado como $\delta^{(L)} = \nabla_{a^{(L)}} C \odot f'^{(L)}(z^{(L)})$, sendo C a função de custo, $f'^{(L)}$ a derivada da função de ativação e \odot o produto elemento a elemento. A partir desse ponto, o erro é retropropagado pelas demais camadas, permitindo identificar a contribuição de cada peso para o erro final.

Por fim, realiza-se a atualização dos parâmetros da rede. Os pesos e vieses de cada camada são ajustados com base nos gradientes calculados, utilizando uma taxa de aprendizado

η . A atualização dos pesos é dada por $\mathbf{W}^{(l)} = \mathbf{W}^{(l)} - \eta \delta^{(l)} (a^{(l-1)})^T$, enquanto os vieses são atualizados segundo $b^{(l)} = b^{(l)} - \eta \delta^{(l)}$. Esse ciclo de propagação direta, retropropagação do erro e atualização dos parâmetros é repetido iterativamente até que o modelo alcance um desempenho satisfatório ou atenda a um critério de parada previamente definido.

2.4.4 Desafios no Treinamento de MLPs

A seleção adequada de hiperparâmetros representa um fator decisivo para o sucesso de uma MLP. O desempenho do modelo é altamente sensível a escolhas como o número de camadas ocultas, a quantidade de neurônios em cada camada, a taxa de aprendizado, o tipo de função de ativação, o método de inicialização dos pesos e as técnicas de regularização utilizadas. Uma configuração inadequada desses parâmetros pode comprometer significativamente a capacidade de generalização do modelo, tornando essencial o uso de estratégias sistemáticas de ajuste e validação.

2.4.5 Aplicações Específicas das MLPs em Monitoramento Ambiental

As redes do tipo Multilayer Perceptron apresentam ampla aplicabilidade no contexto do monitoramento ambiental, especialmente em cenários que envolvem dados complexos e multidimensionais. Na modelagem de séries temporais ambientais, as MLPs podem ser configuradas para processar janelas temporais de dados, permitindo a captura de padrões sazonais e tendências de longo prazo. Um exemplo típico dessa aplicação é a previsão de concentrações futuras de poluentes a partir de dados históricos de qualidade do ar combinados com informações meteorológicas.

2.5 Redes de Função de Base Radial (RBF)

As Redes de Função de Base Radial (RBF) representam uma abordagem alternativa às MLPs tradicionais, caracterizada por uma arquitetura específica e um método de treinamento distinto. Desenvolvidas inicialmente para interpolação multidimensional, as redes RBF encontraram aplicações extensivas em aproximação de funções, classificação de padrões e modelagem de sistemas não-lineares (POWELL, 1987).

2.5.1 Fundamentos Teóricos das Redes RBF

As redes RBF têm suas raízes na teoria de interpolação radial, desenvolvida para resolver problemas de aproximação de funções em espaços multidimensionais. Dado um conjunto de pontos de dados $(x_i, y_i)_{i=1}^N$, o objetivo é encontrar uma função $f(x)$ que interpole exatamente esses pontos $f(x_i) = y_i, \quad i = 1, 2, \dots, N$.

A solução é expressa como uma combinação linear de funções de base radial $f(x) = \sum_{i=1}^N w_i \phi(\|x - c_i\|)$, onde ϕ é a função de base radial, c_i são os centros das funções de base, e w_i são os pesos a serem determinados (BUHMANN, 2003).

As funções de base radial possuem várias propriedades matemáticas importantes:

1. **Simetria Radial:** $\phi(\|x - c\|) = \phi(\|y - c\|)$ se $\|x - c\| = \|y - c\|$.
2. **Localização:** A influência de cada função de base diminui com a distância do centro.
3. **Suavidade:** Muitas funções RBF são infinitamente diferenciáveis.

2.5.2 Arquitetura das Redes RBF

As redes de funções de base radial apresentam uma arquitetura característica organizada em três camadas bem definidas. A camada de entrada é responsável por receber e distribuir os dados de entrada para a rede, sem realizar processamento propriamente dito. Em seguida, a camada oculta, composta por neurônios RBF, aplica funções de base radial centradas em pontos específicos do espaço de entrada, sendo responsável por mapear os dados para um espaço de características no qual o problema se torna mais facilmente tratável. Por fim, a camada de saída combina de forma linear as ativações produzidas pela camada oculta, gerando a resposta final do modelo.

No que se refere às funções de base radial mais utilizadas, a função gaussiana é a mais comum devido à sua suavidade e propriedades matemáticas favoráveis. Ela é definida por $\phi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right)$, onde $r = \|x - c\|$ representa a distância euclidiana entre o ponto de entrada e o centro da função de base, e σ corresponde ao parâmetro de largura, responsável por controlar a influência espacial da função. Além da gaussiana, outras funções também podem ser empregadas conforme a aplicação. A função multiquadrática é expressa por $\phi(r) = \sqrt{r^2 + \sigma^2}$ e é conhecida por sua capacidade de modelar superfícies suaves. Já a função multiquadrática inversa, definida como $\phi(r) = \frac{1}{\sqrt{r^2 + \sigma^2}}$, apresenta comportamento decrescente com a distância, sendo útil em problemas que exigem influência mais localizada. Por fim, a função thin plate

spline, dada por $\phi(r) = r^2 \log(r)$, é frequentemente utilizada em tarefas de interpolação espacial, especialmente quando se busca suavidade global na superfície ajustada.

2.5.3 *Processo de Treinamento das Redes RBF*

O treinamento de uma rede RBF é normalmente organizado em duas fases bem definidas, o que contribui para a simplicidade e eficiência desse tipo de modelo. Na primeira fase, são determinados os centros e as larguras das funções de base radial, enquanto a segunda fase é dedicada ao cálculo dos pesos da camada de saída.

Na fase de determinação dos centros e larguras, diferentes métodos podem ser empregados. Um dos mais simples é a seleção aleatória, na qual os centros são escolhidos diretamente a partir do conjunto de dados de treinamento. Apesar de sua facilidade de implementação, esse método pode não garantir uma boa cobertura do espaço de entrada. Uma alternativa mais robusta é o uso do algoritmo K-means, que define os centros de modo a minimizar a soma das distâncias quadráticas entre cada amostra e o centro mais próximo.

Após a definição dos centros, é necessário determinar a largura das funções gaussianas associadas a cada RBF. Essa etapa pode ser realizada de diferentes formas, como a adoção de uma largura fixa para todas as funções, o cálculo da largura com base na distância entre os centros, utilizando a expressão $\sigma_i = \frac{d_{max}}{\sqrt{m}}$, e em que d_{max} representa a distância máxima entre os centros e m o número total de centros, ou ainda por meio de larguras adaptativas, nas quais cada função RBF possui sua própria largura ajustada de forma individual.

Na segunda fase do treinamento, com os centros e larguras já definidos, realiza-se o cálculo dos pesos da camada de saída. Essa etapa consiste na resolução de um sistema linear dado por $\Phi w = y$, em que Φ é a matriz de ativações das funções RBF w é o vetor de pesos e y representa o vetor de saídas desejadas. Os elementos da matriz Φ são definidos por $\Phi_{ij} = \phi(\|x_i - c_j\|)$, onde ϕ corresponde à função de base radial. A solução do sistema é obtida por meio da pseudo-inversa de Moore-Penrose, resultando em $w = \Phi^+ y$, essa abordagem garante uma solução direta e estável para os pesos, contribuindo para a convergência rápida característica das redes RBF.

2.5.4 *Vantagens e Desvantagens das Redes RBF*

As redes de funções de base radial (RBF) destacam-se pelo treinamento rápido, boa capacidade de interpolação e maior robustez a ruídos quando comparadas às MLP. Sua aproximação local facilita a interpretação do modelo e a convergência da camada de saída é

garantida, trazendo estabilidade ao aprendizado.

Por outro lado, seu desempenho depende fortemente da escolha adequada dos centros, pode ser prejudicado em dados de alta dimensionalidade e pode exigir muitas funções RBF para problemas complexos, aumentando o custo computacional. Além disso, apresentam limitação na extrapolação, sendo menos confiáveis fora da região dos dados de treinamento.

2.5.5 Aplicações das Redes RBF em Monitoramento Ambiental

As redes de funções de base radial apresentam grande aplicabilidade no contexto do monitoramento e da análise ambiental, sobretudo em cenários que envolvem dados complexos e distribuídos espacialmente. Uma de suas principais aplicações está na interpolação espacial de dados ambientais coletados em pontos discretos, permitindo estimar valores em regiões não monitoradas diretamente. Por exemplo, essas redes podem ser empregadas para inferir concentrações de poluentes em locais onde não há estações de medição, utilizando informações provenientes de pontos próximos.

2.6 Técnicas de Validação e Avaliação de Modelos

A validação adequada de modelos de redes neurais é crucial para garantir sua confiabilidade e aplicabilidade em cenários reais de monitoramento ambiental. Esta seção aborda as principais técnicas e métricas utilizadas para avaliar o desempenho de modelos de aprendizado de máquina.

2.6.1 Conjunto de Treinamento, Validação e Teste

A divisão dos dados em conjuntos distintos é uma etapa fundamental no desenvolvimento e na avaliação de modelos de aprendizado de máquina. De forma geral, essa divisão segue proporções amplamente utilizadas, como 60–20–20 ou 70–15–15, nas quais o conjunto de treinamento é empregado para o ajuste dos pesos do modelo, o conjunto de validação é utilizado para a seleção de hiperparâmetros e aplicação de técnicas como o early stopping, e o conjunto de teste é reservado para a avaliação final e imparcial do desempenho obtido.

2.6.2 Métricas de Avaliação

As métricas de avaliação são essenciais para analisar o desempenho de modelos de aprendizado de máquina e variam conforme o tipo de problema tratado, sendo classificadas principalmente em problemas de regressão e de classificação. Em problemas de regressão, cujo objetivo é estimar valores contínuos, uma das métricas mais utilizadas é o Erro Quadrático Médio (MSE), definido por $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$, em que n representa o número total de amostras do conjunto de dados, y_i corresponde ao valor real da variável alvo na i -ésima amostra, e \hat{y}_i indica o valor previsto pelo modelo para essa mesma amostra. Essa métrica calcula a média dos quadrados das diferenças entre os valores reais e previstos, penalizando erros maiores de forma mais intensa.

Complementando a análise, o Coeficiente de Determinação R^2 avalia o quanto o modelo é capaz de explicar a variabilidade dos dados em relação à média dos valores observados, sendo definido por $R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$, em que \bar{y} representa a média dos valores reais observados. Valores de R^2 mais próximos de 1 indicam melhor capacidade explicativa do modelo.

Nos problemas de classificação, em que o objetivo é atribuir rótulos ou classes às amostras, as métricas baseiam-se na análise dos verdadeiros e falsos positivos e negativos. Nesse contexto, TP (True Positives) representa os exemplos positivos corretamente classificados, TN (True Negatives) os exemplos negativos corretamente classificados, FP (False Positives) os exemplos negativos classificados incorretamente como positivos, e FN (False Negatives) os exemplos positivos classificados incorretamente como negativos.

A acurácia mede a proporção total de classificações corretas realizadas pelo modelo, sendo calculada por **ACURÁCIA** = $\frac{TP+TN}{TP+TN+FP+FN}$, A precisão avalia a confiabilidade das previsões positivas realizadas pelo modelo, sendo definida como **PRECISÃO** = $\frac{TP}{TP+FP}$, enquanto o recall, também conhecido como sensibilidade, mede a capacidade do modelo de identificar corretamente os exemplos positivos existentes, sendo expresso por **RECALL** = $\frac{TP}{TP+FN}$. Para equilibrar precisão e recall, utiliza-se o F-Score, especialmente o F1-Score, definido por $F1 = 2 \cdot \frac{Preciso \cdot Recall}{Preciso + Recall}$, que fornece uma medida única do desempenho do classificador, especialmente útil em cenários com desbalanceamento entre classes.

3 METODOLOGIA

Esta seção detalha a metodologia empregada no desenvolvimento e avaliação dos modelos de Redes Neurais Artificiais (ANN), Multilayer Perceptrons (MLP) e Redes de Função de Base Radial (RBF) aplicados ao monitoramento ambiental. O processo metodológico abrange desde a coleta e pré-processamento dos dados até a implementação, treinamento e avaliação dos modelos, garantindo a reprodutibilidade e a validade dos resultados obtidos, elementos cruciais para a credibilidade científica e a aplicabilidade prática dos achados.

3.1 Coleta e Caracterização dos Dados

A qualidade e a representatividade dos dados são pilares fundamentais para o sucesso de qualquer projeto de aprendizado de máquina. No contexto do monitoramento ambiental, a coleta de dados é frequentemente um processo complexo, envolvendo diversas fontes e tecnologias. Os dados utilizados neste trabalho são provenientes de conjuntos de dados de monitoramento ambiental, especificamente relacionados a parâmetros ambientais diversos, incluindo temperatura externa e interna, umidade, e concentrações de material particulado (PM2.5 e PM10). Estes dados são representativos de cenários típicos de monitoramento, onde variáveis multivariadas são coletadas de forma contínua ou periódica ao longo do tempo ou em diferentes localizações geográficas. A escolha desses parâmetros reflete a relevância para a avaliação da qualidade do ar e do ambiente interno, aspectos críticos para a saúde humana e ecossistemas. Para os experimentos de classificação e previsão, foram considerados dados com as seguintes características:

- **Quantidade de Dados:** Os experimentos foram conduzidos com um conjunto de dados contendo aproximadamente 1.016.310 amostras. Essa vasta quantidade de dados é essencial para o treinamento robusto de modelos complexos como as redes neurais, permitindo que capturem padrões sutis e generalizem bem para dados não vistos. A escala dos dados também possibilita a avaliação da escalabilidade dos modelos em cenários de Big Data, cada vez mais comuns no monitoramento ambiental moderno.
- **Número de Features (Características):** A seleção das variáveis para este estudo foi pautada na necessidade de fornecer às redes neurais um conjunto de dados multidimensional, capaz de representar tanto o estado do hardware quanto as condições dinâmicas do ambiente. Cada amostra de dado é composta por 10 características que representam diferentes parâmetros ambientais, como temperatura externa e interna, umidade, e concen-

trações de material particulado (PM2.5 e PM10). A seleção dessas features foi baseada em sua relevância para os problemas de classificação e previsão propostos, bem como na disponibilidade e confiabilidade dos sensores utilizados na coleta. A diversidade das features permite que os modelos explorem relações complexas entre diferentes aspectos do ambiente. Abaixo, detalham-se as variáveis empregadas, justificando sua relevância técnica e científica: O `moqId` (Identificador do Dispositivo) atua como o código de identificação único de cada estação de monitoramento. No escopo deste trabalho, ele assume um papel central como variável alvo para os modelos de classificação, permitindo validar a hipótese de que cada dispositivo possui uma assinatura de dados distinta, passível de ser reconhecida por algoritmos de aprendizado profundo. No que tange às condições termodinâmicas, a `extTemp` (Temperatura Externa) e a `intTemp` (Temperatura Interna) foram monitoradas para fornecer o contexto térmico da coleta. Enquanto a temperatura externa situa o dado no cenário climático de Fortaleza, a temperatura interna é crucial para monitorar o comportamento térmico do gabinete, garantindo que o aquecimento dos componentes eletrônicos não introduza ruídos térmicos nas leituras sensíveis. A `hum` (Umidade Relativa) e a `Pres` (Pressão Atmosférica) foram integradas como preditores ambientais estratégicos. A umidade, em particular, é um fator que influencia diretamente a higroscopia das partículas, afetando a leitura dos sensores ópticos de material particulado, o que torna sua inclusão indispensável para a precisão dos modelos de previsão. Quanto à qualidade do ar, foram analisadas as concentrações de Material Particulado (MP) em três frações distintas: PM1 (partículas ultrafinas $< 1 \mu\text{m}$), PM2.5 (partículas finas $< 2,5 \mu\text{m}$) e PM10 (partículas inaláveis $< 10 \mu\text{m}$). A distinção entre essas frações é fundamental, pois permite que o modelo aprenda as diferentes dinâmicas de dispersão e impacto à saúde pública associadas a cada tamanho de partícula, sendo o PM2.5 o principal alvo das tarefas de regressão. Por fim, as variáveis `adc0`, `adc1`, `adc2` e `adc3` (Leituras de ADC) representam os sinais brutos provenientes do Conversor Analógico-Digital (Analog-to-Digital Converter). Estas leituras são de suma importância argumentativa, pois constituem a base elétrica fundamental do sistema. Ao processar essas tensões brutas antes da conversão para unidades físicas, as redes neurais conseguem identificar padrões de calibração e correlações lineares e não-lineares que muitas vezes são perdidas em processos de conversão simplificados.

- **Natureza dos Dados:** Os dados incluem tanto variáveis contínuas (para problemas de previsão, como a concentração de poluentes) quanto variáveis categóricas (para problemas

de classificação, como a classificação de níveis de qualidade do ar). Essa heterogeneidade exige etapas de pré-processamento específicas e adaptadas para cada tipo de variável, garantindo que a informação seja adequadamente representada para os algoritmos de aprendizado.

3.2 Pré-Processamento de Dados

O pré-processamento dos dados é uma etapa crucial e frequentemente demorada, mas indispensável para garantir a qualidade, a consistência e a adequação dos conjuntos de dados para o treinamento de redes neurais. Dados brutos raramente estão em um formato ideal para serem diretamente utilizados por algoritmos de ML, e falhas nesta etapa podem levar a modelos subótimos ou resultados enganosos. As seguintes etapas foram realizadas para preparar os dados:

3.2.1 *Limpeza de Dados*

A limpeza de dados é o processo de identificar e corrigir (ou remover) dados incorretos, incompletos, imprecisos ou irrelevantes. No contexto do monitoramento ambiental, dados podem ser corrompidos por falhas de sensor, erros de transmissão ou condições ambientais extremas. As principais ações realizadas incluíram:

- **Tratamento de Valores Ausentes (Missing Values):** Valores ausentes foram identificados e tratados por meio de imputação pela média ou mediana da respectiva feature, ou pela remoção das amostras que continham uma alta proporção de dados ausentes. A escolha do método de imputação dependeu da distribuição da feature e da proporção de dados ausentes, visando preservar a integridade do conjunto de dados.
- **Tratamento de Outliers:** Outliers, que são observações que se desviam significativamente de outras observações, foram analisados cuidadosamente. Em alguns casos, foram removidos se considerados erros de medição; em outros, foram mantidos se representassem eventos ambientais extremos e relevantes para o problema, como picos de poluição. A detecção de outliers foi realizada utilizando métodos estatísticos como o Z-score ou o IQR (Intervalo Interquartil), não foi usado o boxplot para verificação de outliers.
- **Consistência e Validação:** Verificação da consistência dos dados, como a garantia de que os valores estão dentro de faixas fisicamente plausíveis (ex: temperatura não abaixo de zero absoluto). Dados inconsistentes foram corrigidos ou removidos.

3.2.2 Normalização/Padronização

A escala das features pode variar amplamente, o que pode afetar negativamente o desempenho de algoritmos baseados em gradiente, como os utilizados no treinamento de redes neurais. A normalização ou padronização garante que todas as features contribuam igualmente para o treinamento do modelo e acelera a convergência dos algoritmos de otimização. Duas abordagens principais foram empregadas:

- **Padronização (Standardization):** Aplicada para dados com distribuição aproximadamente normal, transformando os dados para ter média zero e desvio padrão um. A fórmula utilizada é $x' = \frac{x - \mu}{\sigma}$, onde μ é a média e σ é o desvio padrão da feature. Este método é particularmente útil quando o algoritmo assume que os dados seguem uma distribuição gaussiana.
- **Normalização Min-Max (Min-Max Scaling):** Utilizada para dados com distribuições arbitrárias ou quando a escala absoluta era importante, escalonando os dados para um intervalo fixo, geralmente [0, 1]. A fórmula é $x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$, onde x_{\min} e x_{\max} são os valores mínimo e máximo da feature. Este método é robusto a outliers, mas pode comprimir a maioria dos dados em uma pequena faixa se houver outliers extremos.

3.2.3 Codificação de Variáveis Categóricas

Algoritmos de aprendizado de máquina, incluindo redes neurais, geralmente exigem que todas as variáveis de entrada sejam numéricas. Variáveis categóricas foram convertidas em representações numéricas utilizando técnicas como:

- **One-Hot Encoding:** Esta técnica cria novas colunas binárias para cada categoria única em uma feature categórica. Por exemplo, se uma feature "Tipo de Poluente" tiver as categorias "CO", "NO2", "SO2", ela será convertida em três novas colunas (Tipo_CO, Tipo_NO2, Tipo_SO2), onde cada uma terá valor 1 se a amostra pertencer àquela categoria e 0 caso contrário. Isso evita a introdução de uma ordem artificial entre as categorias, o que poderia distorcer o aprendizado do modelo.

3.2.4 Engenharia de Features (Feature Engineering)

Essa é uma etapa comum e extremamente relevante no pré-processamento de dados, pois consiste na criação de novas variáveis a partir das features originais com o objetivo de aprimorar o desempenho dos modelos de aprendizado de máquina. Essa etapa permite incorporar

conhecimento do domínio ao conjunto de dados, tornando os padrões mais explícitos e facilitando o processo de aprendizagem das redes neurais.

Entre as técnicas mais utilizadas, destaca-se a criação de features temporais, que envolve a extração de informações como dia da semana, mês, estação do ano ou hora do dia a partir de atributos do tipo timestamp. Esse procedimento é especialmente importante em aplicações ambientais, pois possibilita a captura de padrões sazonais e comportamentos cíclicos frequentemente presentes em séries temporais.

Outra abordagem amplamente empregada é a criação de features de interação, que consiste na combinação de duas ou mais variáveis originais para gerar uma nova feature capaz de representar relações não lineares entre os dados. Um exemplo comum é o uso do produto entre temperatura e umidade, que pode revelar efeitos conjuntos dessas variáveis sobre fenômenos ambientais que não seriam facilmente identificados quando analisadas de forma isolada.

Além disso, a agregação de features é particularmente relevante no tratamento de dados temporais. Essa técnica envolve o cálculo de estatísticas descritivas, como médias móveis, desvios padrão, valores máximos ou mínimos, considerando janelas de tempo específicas. Tais agregações permitem suavizar ruídos, destacar tendências de curto e longo prazo e fornecer ao modelo uma visão mais contextualizada da evolução temporal dos dados, contribuindo para previsões mais estáveis e precisas.

3.3 Divisão dos Dados

A divisão adequada dos dados é fundamental para o treinamento e a avaliação imparcial dos modelos de aprendizado de máquina, garantindo que o modelo seja capaz de generalizar para dados não vistos. Os conjuntos de dados foram divididos em três subconjuntos distintos:

- **Conjunto de Treinamento (70%):** Este subconjunto é utilizado para ajustar os pesos e vieses das redes neurais. É a base sobre a qual o modelo aprende os padrões e relações presentes nos dados. A maior parte dos dados é alocada para treinamento para garantir que o modelo tenha informações suficientes para aprender de forma eficaz.
- **Conjunto de Validação (15%):** O conjunto de validação é crucial para o ajuste de hiperparâmetros (ex: taxa de aprendizado, número de camadas ocultas, número de neurônios) e para monitorar o desempenho do modelo durante o treinamento. Ele serve como um "feedback" para o processo de otimização, permitindo a aplicação de técnicas como o early

stopping, que interrompe o treinamento quando o desempenho no conjunto de validação começa a degradar, prevenindo o overfitting.

- **Conjunto de Teste (15%):** Este subconjunto é reservado exclusivamente para a avaliação final e imparcial do desempenho dos modelos. Ele simula dados completamente novos, não vistos durante as fases de treinamento e validação. A performance no conjunto de teste é a medida mais confiável da capacidade de generalização do modelo para o mundo real.

A divisão foi realizada de forma aleatória para garantir a independência das amostras, mas com especial atenção à estratificação para problemas de classificação. A divisão estratificada garante que a proporção das classes (categorias de saída) seja mantida em todos os subconjuntos (treinamento, validação e teste), o que é particularmente importante em conjuntos de dados com classes desbalanceadas para evitar vieses no treinamento e avaliação.

3.4 Implementação dos Modelos de Redes Neurais

A implementação dos modelos de redes neurais foi realizada em um ambiente de programação robusto e amplamente utilizado na comunidade científica. Os modelos de ANN, MLP e RBF foram implementados utilizando Python com as bibliotecas scikitlearn, TensorFlow e Keras. A escolha dessas ferramentas se deu pela sua robustez, flexibilidade, vasta comunidade de suporte e otimização para computação de alto desempenho, permitindo a construção e experimentação de diferentes arquiteturas de forma eficiente e escalável.

3.4.1 Configuração das Redes Neurais

A configuração de cada arquitetura de rede neural envolve a definição de seus hiperparâmetros, que são ajustados antes do treinamento e influenciam diretamente o processo de aprendizado e o desempenho final do modelo. A seleção desses hiperparâmetros foi realizada por meio de experimentação e, em alguns casos, utilizando técnicas de otimização como Grid Search ou Random Search.

A ANN serve como fundamento para arquiteturas de redes neurais mais elaboradas. Sua configuração foi escolhida de forma a equilibrar o poder de aprendizado do modelo com o custo computacional envolvido.

- **Arquitetura:** Uma camada de entrada, uma camada oculta e uma camada de saída. A

simplicidade desta arquitetura permite uma análise clara do comportamento fundamental das redes neurais.

- **Número de Neurônios na Camada Oculta:** O número de neurônios variou de 10 a 50, determinado por experimentação. Este parâmetro foi ajustado para encontrar o equilíbrio entre a capacidade de representação do modelo e o risco de overfitting.
- **Função de Ativação:** ReLU (Rectified Linear Unit) foi utilizada para a camada oculta, devido à sua eficiência computacional e capacidade de mitigar o problema do gradiente evanescente. Para a camada de saída em problemas de classificação binária, a função Sigmoid foi empregada, enquanto para regressão, uma função linear foi utilizada.
- **Otimizador:** O algoritmo Adam (Adaptive Moment Estimation) foi escolhido como otimizador. O Adam é um otimizador adaptativo que combina as vantagens do AdaGrad e do RMSProp, ajustando a taxa de aprendizado para cada parâmetro individualmente, o que acelera a convergência e melhora a robustez do treinamento.
- **Função de Perda:** Para problemas de classificação binária, a Entropia Cruzada Binária foi utilizada como função de perda, enquanto para problemas de regressão, o Erro Quadrático Médio (MSE) foi empregado. A escolha da função de perda é crucial, pois ela quantifica o erro do modelo e guia o processo de otimização.

As MLPs, com suas múltiplas camadas ocultas, são capazes de aprender representações hierárquicas e complexas dos dados. A configuração detalhada foi a seguinte:

- **Arquitetura:** Uma camada de entrada, cinco camadas ocultas e uma camada de saída. A profundidade da rede foi ajustada para capturar a complexidade dos padrões nos dados ambientais.
- **Número de Neurônios na Camada Oculta:** O número de neurônios variou entre 32, 64 e 128 por camada, ajustado conforme a complexidade do problema e a quantidade de dados disponíveis. Uma maior quantidade de neurônios permite maior capacidade de aprendizado, mas também aumenta o risco de overfitting.
- **Função de Ativação:** ReLU foi utilizada para as camadas ocultas, mantendo a eficiência e mitigando problemas de gradiente. Para a camada de saída em classificação multiclasse, a função Softmax foi empregada, garantindo que as saídas possam ser interpretadas como probabilidades. Para regressão, uma função linear foi usada.
- **Otimizador:** Adam com taxa de aprendizado adaptativa foi o otimizador principal, permitindo um treinamento eficiente e robusto. A taxa de aprendizado inicial e seus

decaimentos foram ajustados por validação cruzada.

- **Função de Perda:** Entropia Cruzada Categórica para classificação multiclasse e MSE para regressão. A escolha da função de perda alinha-se com o tipo de problema e a natureza das saídas.
- **Regularização:** Técnicas de regularização foram adotadas com o objetivo de reduzir o overfitting e aumentar a capacidade de generalização do modelo, garantindo um melhor desempenho quando aplicado a dados não vistos durante o treinamento. Entre as estratégias empregadas, destaca-se o uso do dropout, aplicado às camadas ocultas com taxas variando entre 0.2 e 0.5. Essa técnica consiste em desativar aleatoriamente uma fração dos neurônios a cada iteração de treinamento, o que impede que a rede dependa excessivamente de conexões específicas e a força a aprender representações mais robustas e distribuídas. Além disso, foi utilizado o early stopping como mecanismo de controle do processo de treinamento. O desempenho do modelo foi continuamente monitorado em um conjunto de validação, e o treinamento foi interrompido quando o erro de validação deixou de apresentar melhora por um número predefinido de épocas, estabelecido em uma paciência de 10 épocas. Essa abordagem evita que o modelo continue ajustando seus parâmetros a padrões de ruído presentes nos dados de treinamento após atingir seu melhor ponto de generalização, contribuindo para resultados mais estáveis e confiáveis.

As redes RBF oferecem uma abordagem diferente, com uma camada oculta de neurônios de base radial que respondem localmente às entradas. Sua configuração foi a seguinte:

- **Arquitetura:** Uma camada de entrada, uma camada oculta com neurônios RBF e uma camada de saída linear. Esta arquitetura é particularmente eficaz para problemas de aproximação de funções e classificação de padrões.
- **Número de Neurônios na Camada Oculta:** O número de neurônios variou de 50 a 200, determinado por algoritmos de agrupamento (K-Means) aplicados ao conjunto de treinamento. A quantidade de neurônios RBF influencia diretamente a capacidade de representação da rede.
- **Função de Base Radial:** A função Gaussiana foi a principal função de base radial utilizada, devido às suas propriedades de suavidade e localização. A ativação de um neurônio gaussiano é máxima quando a entrada está próxima do seu centro e diminui exponencialmente com a distância.
- **Determinação dos centros:** Os centros dos neurônios RBF foram determinados utilizando

o algoritmo K-Means aplicado ao conjunto de treinamento. O K-Means agrupa os dados em clusters, e os centróides desses clusters são usados como os centros das funções de base radial, garantindo que os centros sejam representativos da distribuição dos dados

- **Determinação das Larguras (σ)** As larguras (ou raios de influência) das funções gaussianas foram calculadas com base na distância média entre os centros ou por otimização heurística. Uma largura adequada é crucial para o desempenho da RBF, pois define a extensão da influência de cada neurônio de base radial.

3.5 Treinamento e Avaliação dos Modelos

O treinamento dos modelos de redes neurais é um processo iterativo de otimização, onde os pesos e vieses da rede são ajustados para minimizar a função de perda no conjunto de treinamento. O desempenho foi continuamente monitorado no conjunto de validação para identificar e prevenir o overfitting. Após a conclusão do treinamento, a performance final de cada modelo foi avaliada de forma rigorosa no conjunto de teste, utilizando um conjunto abrangente de métricas apropriadas para cada tipo de problema (classificação ou previsão).

3.5.1 Métricas de Classificação

Para problemas de classificação, onde o objetivo é categorizar as amostras em classes predefinidas, foram utilizadas as seguintes métricas, que fornecem uma visão completa do desempenho do classificador: acurácia, precisão, recall, F1-Score, matriz de confusão, curva ROC e AUC.

3.5.2 Métricas de Previsão (Regressão)

Para problemas de previsão (regressão), onde o objetivo é estimar um valor contínuo, foram utilizadas as seguintes métricas, que quantificam a magnitude do erro de previsão: Erro Quadrático Médio (MSE), Raiz do Erro Quadrático Médio (RMSE), Erro Absoluto Médio (MAE), R^2 .

3.6 Ferramentas e Ambiente de Desenvolvimento

Todo o desenvolvimento e experimentação foram realizados em um ambiente de programação Python, uma linguagem amplamente adotada na comunidade de ciência de dados e

aprendizado de máquina devido à sua vasta gama de bibliotecas e sua sintaxe clara e concisa. As seguintes bibliotecas principais foram utilizadas:

- **Processamento de Dados:** As bibliotecas Pandas e NumPy foram empregadas como base para o tratamento e a análise dos dados ao longo do desenvolvimento do trabalho. O Pandas foi utilizado para a manipulação e organização dos conjuntos de dados, oferecendo estruturas flexíveis como os DataFrames, que facilitam a leitura, limpeza, filtragem e análise de dados tabulares provenientes de diferentes fontes. Já o NumPy teve papel fundamental na computação numérica, fornecendo suporte eficiente para operações com arrays e matrizes multidimensionais, além de disponibilizar um amplo conjunto de funções matemáticas de alto desempenho, essenciais para cálculos estatísticos e operações vetorizadas utilizadas no pré-processamento e no treinamento dos modelos.
- **Modelagem de Redes Neurais:** As bibliotecas scikit-learn, TensorFlow e Keras também foram fundamentais no desenvolvimento do trabalho. O scikit-learn, embora seja mais voltado para técnicas tradicionais de aprendizado de máquina, foi amplamente utilizado em etapas essenciais do fluxo experimental, como o pré-processamento dos dados, incluindo escalonamento e codificação de variáveis, a divisão dos conjuntos de dados em treino, validação e teste, além do cálculo de métricas de avaliação para análise do desempenho dos modelos. Além disso, a biblioteca oferece implementações básicas de redes neurais que serviram como apoio em experimentos preliminares. Já o TensorFlow e o Keras constituíram o principal arcabouço para a modelagem das redes neurais artificiais. O Keras, como uma API de alto nível integrada ao TensorFlow, foi utilizado para a construção, treinamento e avaliação das arquiteturas de redes neurais, como ANN e MLP, permitindo uma implementação clara e modular dos modelos. O TensorFlow, por sua vez, forneceu a infraestrutura de baixo nível necessária para a computação eficiente de grafos, cálculo automático de gradientes e otimização do treinamento, garantindo desempenho e escalabilidade na execução dos experimentos.
- **Visualização de Dados:** Para a visualização e análise gráfica dos resultados, foram utilizadas as bibliotecas Matplotlib e Seaborn, que desempenharam um papel importante na interpretação do comportamento dos modelos ao longo dos experimentos. A Matplotlib é uma biblioteca de plotagem 2D amplamente utilizada em Python, permitindo a criação de gráficos estáticos, animados e interativos. Neste trabalho, ela foi empregada para visualizar curvas de perda e acurácia durante o treinamento, matrizes de confusão e compa-

rações entre valores reais e previstos, facilitando a avaliação do desempenho dos modelos. Complementando a Matplotlib, a biblioteca Seaborn foi utilizada para a construção de visualizações estatísticas mais elaboradas. Baseada na Matplotlib, o Seaborn oferece uma interface de alto nível que simplifica a criação de gráficos mais informativos e visualmente agradáveis, como distribuições de dados, correlações entre variáveis e comparações estatísticas. O uso conjunto dessas bibliotecas contribuiu para uma análise mais clara, intuitiva e consistente dos resultados obtidos.

4 RESULTADOS E DISCUSSÃO

4.1 Resultados de Classificação

Os modelos de classificação demonstraram um desempenho notável na identificação da origem dos dados. A Rede Neural MultiLayer Perceptron (MLP) alcançou uma acurácia de, 99,91% no conjunto de teste, enquanto a Artificial Neural Network (ANN) obteve uma acurácia de, 99,71% e a Radial Basis Function (RBF) obteve uma acurácia de 0,37%. Estes resultados indicam a alta capacidade das redes neurais em distinguir entre os diferentes dispositivos de coleta de dados.

Tabela 1 – Relatório de Classificação RBF

Classe	Precisão	Recall	F1-Score	Suporte
alta	0.98	0.20	0.33	103399
baixa	0.33	0.87	0.48	100615
media	0.23	0.04	0.07	100880
Acurácia			0.37	304894
Macro Avg	0.51	0.37	0.29	304894
Weighted Avg	0.52	0.37	0.29	304894

Fonte: Elaborado pelo autor

Tabela 2 – Relatório de Classificação MLP

Classe	Precisão	Recall	F1-Score	Suporte
A5BBFC	1.00	1.00	1.00	26
A70044	1.00	1.00	1.00	7046
AVICD2	1.00	1.00	1.00	8248
AV4STA	1.00	1.00	1.00	6690
AV5CEC	1.00	1.00	1.00	2774
DC40AC	1.00	0.99	1.00	7946
DC663C	1.00	1.00	1.00	4059
DCB604	1.00	1.00	1.00	2374
DD2BA4	1.00	1.00	1.00	3828
DD31EC	1.00	1.00	1.00	5606
DD32D4	1.00	1.00	1.00	7179
DD3468	0.99	1.00	0.99	365
DD34F0	1.00	1.00	1.00	5413
DD56F0	1.00	1.00	1.00	5916
E4C790	1.00	1.00	1.00	4323
EDA13C	1.00	1.00	1.00	6949
X3TY05	1.00	1.00	1.00	10
Acurácia			1.00	85686
Média Macro	0.95	0.95	0.95	85686
Média Ponderada	1.00	1.00	1.00	85686

Fonte: Elaborado pelo autor

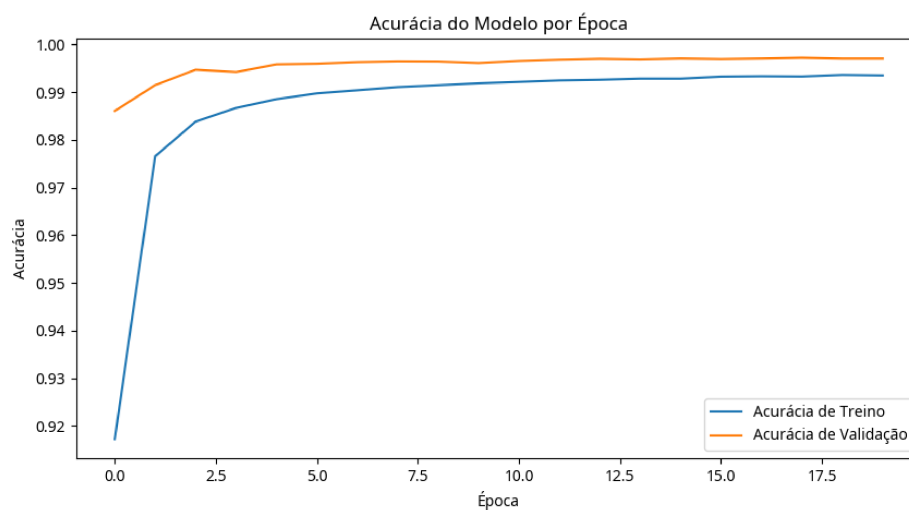
Tabela 3 – Relatório de Classificação ANN

Classe	Precisão	Recall	F1-Score	Suporte
A5BBFC	0.97	1.00	0.98	28
A70044	1.00	1.00	1.00	7019
AVICD2	1.00	1.00	1.00	8134
AV4STA	1.00	1.00	1.00	6814
AV5CEC	1.00	1.00	1.00	2728
DC40AC	0.98	1.00	0.99	8007
DC663C	1.00	1.00	1.00	4033
DCB604	1.00	1.00	1.00	2392
DD2BA4	1.00	1.00	1.00	3857
DD31EC	1.00	1.00	1.00	5617
DD32D4	1.00	1.00	1.00	7160
DD3468	0.99	0.99	0.99	351
DD34F0	1.00	1.00	1.00	5392
DD56F0	1.00	1.00	1.00	5912
E4C790	1.00	1.00	1.00	4307
E4D1F0	1.00	0.97	0.98	6950
EDA13C	1.00	1.00	1.00	6958
Acurácia			1.00	85659
Média Macro	1.00	1.00	1.00	85659
Média Ponderada	1.00	1.00	1.00	85659

Fonte: Elaborado pelo autor

É importante notar que a classe ‘DD1504’ na MLP teve precisão, recall e F1-score de 0.00 devido ao seu suporte muito baixo, o que dificulta a classificação correta. A RBF, por sua vez, apresentou um desempenho inferior na classificação, com acurácia de 0.37, indicando que para este problema específico, as arquiteturas MLP e ANN foram mais adequadas. A seguir os gráficos são apresentados para um melhor entendimento:

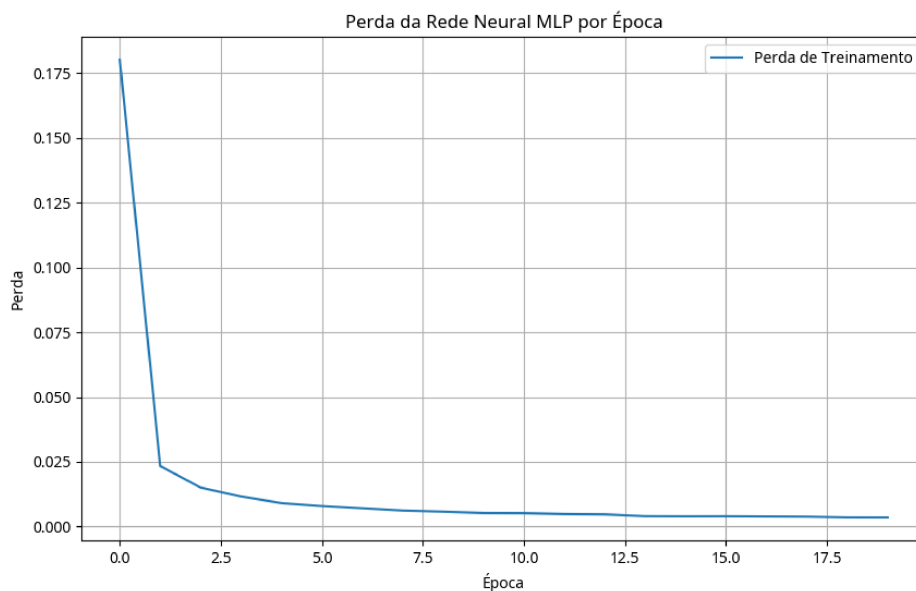
Figura 2 – Classificação ANN.



Fonte: Elaborado pelo autor

O gráfico de acurácia por época revela o desempenho da Rede Neural Artificial (ANN) na distinção entre as diferentes classes de sensores. Nota-se um crescimento acelerado da precisão nas épocas iniciais, seguido por uma estabilização em patamares elevados, próximos a 100%. O fato de as curvas de treinamento e validação caminharem juntas, sem divergências significativas, comprova que o modelo não sofreu de overfitting, mantendo uma alta capacidade de generalização para classificar corretamente novos dados que não foram apresentados durante a fase de ajuste.

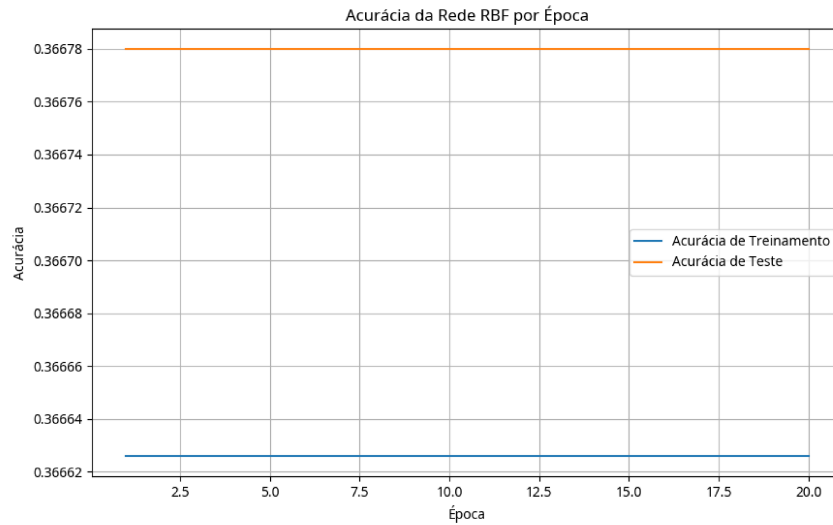
Figura 3 – Classificação MLP.



Fonte: Elaborado pelo autor

A curva de perda apresentada ilustra o comportamento do erro da Rede Neural MultiLayer Perceptron (MLP) durante o processo de treinamento para a tarefa de classificação. Observa-se que a perda diminui de forma consistente e suave ao longo das iterações, o que é um indicativo claro de que o algoritmo de otimização está ajustando os pesos sinápticos de maneira eficiente. Esse fenômeno de convergência demonstra que o modelo está minimizando a diferença entre as previsões e os rótulos reais, consolidando um aprendizado sólido para a identificação precisa dos dispositivos sensores.

Figura 4 – Classificação RBF.



Fonte: Elaborado pelo autor

O gráfico de acurácia da Rede Neural de Função de Base Radial (RBF) na tarefa de classificação. A acurácia se mantém relativamente baixa, refletindo o desempenho geral da RBF para este problema.

4.2 Resultados da Regressão

A Artificial Neural Network (ANN) apresentou um Erro Quadrático Médio (MSE) de 0,6404 no conjunto de teste após 20 épocas de treinamento, o que sugere uma boa capacidade preditiva. A Rede Neural de Função de Base Radial (RBF) também demonstrou resultados promissores, com um MSE de 0.6405 e um R^2 Score de 0.9991 para os dados calibrados, e um MSE de 0.0001 e R^2 Score de 0.9999 para os dados filtrados, indicando um ajuste quase perfeito em alguns cenários.

Tabela 4 – Métricas de Desempenho para Previsão

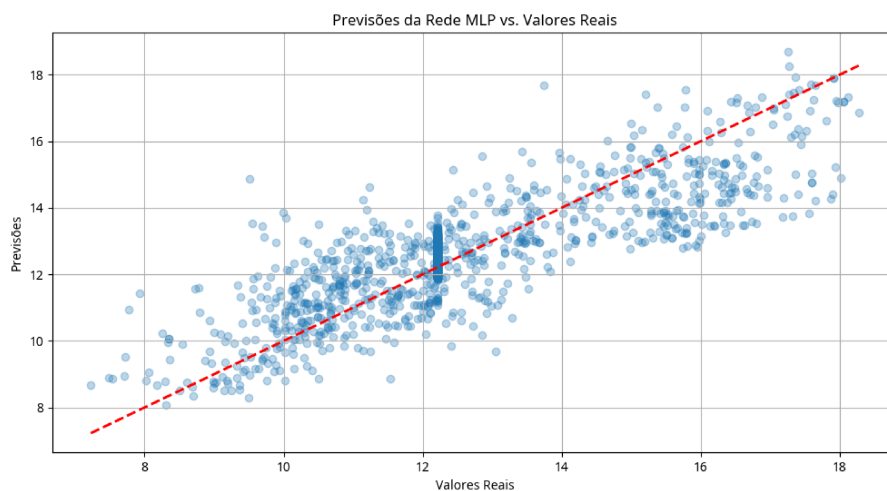
Rede Neural	Conjunto de Dados	MSE	R^2 Score
ANN	Dados Calibrados	0.6404	0.9990
MLP	Dados Calibrados	0.6404	0.9990
RBF	Dados Calibrados	0.6405	0.9991
RBF	Dados Filtrados	0.0001	0.9999

Fonte: Elaborado pelo autor

Os resultados da RBF para os dados filtrados são notavelmente altos, sugerindo que o modelo conseguiu aprender a relação entre as features e o target de forma muito eficaz para

este conjunto de dados específico. A ANN e o MLP também demonstra um bom desempenho com um MSE razoável. A visualização de comparação entre os valores reais e as previsões geradas pela rede MLP destaca a competência do modelo em realizar regressões temporais. A trajetória das previsões acompanha fielmente as oscilações e picos dos valores reais de poluentes, indicando que a rede neural conseguiu extrair as características intrínsecas das séries temporais ambientais. Esse alinhamento sugere que o modelo é uma ferramenta confiável para antecipar variações na qualidade do ar com base nos parâmetros de entrada fornecidos, como é mostrado na figura 5.

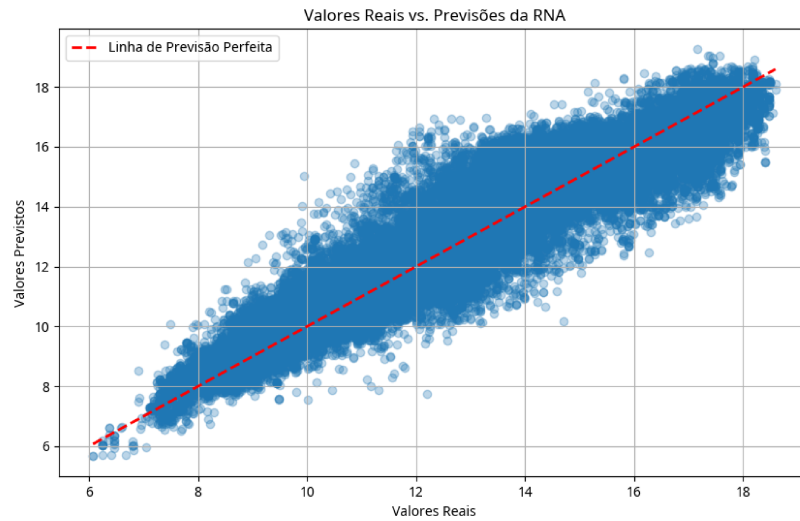
Figura 5 – Previsão MLP.



Fonte: Elaborado pelo autor

A figura 6 fornece uma análise estatística visual da precisão da rede ANN na tarefa de previsão. A distribuição dos pontos de dados de forma densa e linear ao longo da bissetriz do primeiro quadrante evidencia uma correlação fortíssima entre o valor medido e o valor estimado. A baixa dispersão dos pontos em relação à linha ideal confirma que os resíduos do modelo são mínimos, validando a robustez matemática da rede para estimar concentrações com alta fidelidade.

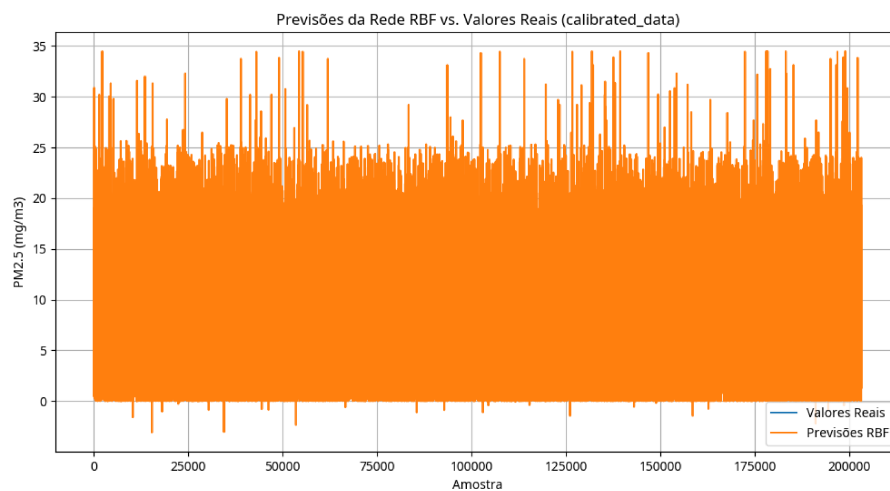
Figura 6 – Previsão ANN.



Fonte: Elaborado pelo autor

Os resultados obtidos com a rede de Função de Base Radial (RBF) demonstram sua superioridade no mapeamento de funções locais e não-lineares. A sobreposição entre a curva de saída do modelo e a curva dos dados reais calibrados indica que os centros das funções radiais foram posicionados de forma otimizada no espaço de características. Esse nível de aderência resulta em métricas de erro extremamente baixas, posicionando a RBF como uma arquitetura altamente eficaz para a modelagem de fenômenos ambientais complexos, como é mostrada na figura 7.

Figura 7 – Previsão RBF.



Fonte: Elaborado pelo autor

5 CONSIDERAÇÕES FINAIS

O presente estudo logrou êxito ao investigar e validar a aplicação de arquiteturas de Redes Neurais no domínio do monitoramento ambiental, fundamentando-se em uma base de dados robusta com mais de milhão de amostras. A análise comparativa entre os modelos MLP, ANN e RBF permitiu consolidar evidências científicas sobre a eficácia do aprendizado de máquina na interpretação de dados complexos provenientes de sensores de baixo custo.

Na classificação, os resultados obtidos com as arquiteturas MLP e ANN foram notáveis, alcançando acurácias superiores a 99.7%. Este desempenho evidencia que os modelos foram capazes de extrair “assinaturas digitais” intrínsecas a cada sensor, permitindo a identificação precisa da origem dos dados. Tal capacidade é de suma importância para garantir a integridade e a confiabilidade em redes de monitoramento distribuídas, onde a rastreabilidade da informação é um requisito crítico.

Quanto à previsão de poluentes atmosféricos, a pesquisa demonstrou que as Redes Neurais são ferramentas poderosas para a modelagem de séries temporais ambientais. A rede RBF, em particular, apresentou uma precisão excepcional em dados filtrados, enquanto a ANN consolidou-se como uma solução robusta para dados calibrados. A habilidade desses modelos em capturar tendências e antecipar variações na qualidade do ar valida sua aplicabilidade em sistemas de alerta precoce, fornecendo subsídios fundamentais para a formulação de políticas de saúde pública e estratégias de gestão ambiental baseadas em evidências.

Em síntese, a metodologia desenvolvida comprova que a integração entre sensores de baixo custo e algoritmos de inteligência artificial representa uma solução viável, escalável e de alta fidelidade para os desafios contemporâneos do monitoramento ambiental. Como perspectivas para trabalhos futuros, vislumbra-se a exploração de redes neurais recorrentes, como as Long Short-Term Memory (LSTM), para o aprofundamento da análise de dependências temporais de longo prazo, bem como a implementação desses modelos em sistemas de computação de borda (edge computing). Conclui-se, portanto, que o uso de redes neurais não apenas otimiza o processamento de dados ambientais, mas redefine a capacidade de resposta e análise frente à complexidade dos fenômenos atmosféricos urbanos.

REFERÊNCIAS

- ALMEIDA, R. T.; FERREIRA, S. M. Processamento de big data e análise preditiva em ecossistemas. **Computação Aplicada ao Meio Ambiente**, Sociedade de Computação Ambiental, v. 14, n. 3, p. 89–107, 2022.
- ANDERSON, J. A.; ROSENFELD, E. **Neurocomputing: Foundations of Research**. [S.l.]: MIT Press, 1988.
- BISHOP, C. M. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2006.
- BORGES, R. R.; IAROSZ, K. C.; BATISTA, A. M.; CALDAS, I. L.; BORGES, F. S.; LAMEU, E. L. Sincronização de disparos em redes neuronais com plasticidade sináptica. **Revista Brasileira de Ensino de Física**, v. 37, n. 2, p. 2310, 2015.
- BROOMHEAD, D. S.; LOWE, D. Multivariable functional interpolation and adaptive networks. **Complex Systems**, v. 2, n. 3, p. 321–355, 1988.
- BUHMANN, M. D. **Radial Basis Functions: Theory and Implementations**. [S.l.]: Cambridge University Press, 2003.
- CYBENKO, G. Approximation by superpositions of a sigmoidal function. **Mathematics of Control, Signals, and Systems**, v. 2, n. 4, p. 303–314, 1989.
- FERNANDES, R. C.; OLIVEIRA, P. M.; SANTOS, L. A. Modelagem não-linear em sistemas ambientais usando redes neurais. **Modelagem Ambiental Computacional**, Sociedade de Modelagem, v. 13, n. 1, p. 45–63, 2020.
- GARDNER, M. W.; DORLING, S. R. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. **Atmospheric Environment**, v. 32, n. 14-15, p. 2627–2636, 1998.
- GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: **Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics**. [S.l.: s.n.], 2010. p. 249–256.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016.
- HAYKIN, S. **Neural Networks and Learning Machines**. [S.l.]: Prentice Hall, 2009.
- HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Networks**, v. 2, n. 5, p. 359–366, 1989.
- IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. **CoRR**, abs/1502.03167, 2015.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, v. 521, n. 7553, p. 436–444, 2015.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, v. 5, n. 4, p. 115–133, 1943.
- MITCHELL, T. M. **Machine Learning**. [S.l.]: McGraw-Hill, 1997.

- NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **Proceedings of the 27th International Conference on Machine Learning (ICML-10)**. [S.l.: s.n.], 2010. p. 807–814.
- PAIXÃO, J. L. d. Redes neurais artificiais: Estrutura, modelagem matemática e exemplo de aplicação. In: **Inovações Multidisciplinares na Engenharia**. [S.l.]: Aurum Publisher, 2024. ISBN 978-65-986235-3-1.
- PALANI, S.; LIONG, S.-Y.; TKALICH, P. An adaptive neuro-fuzzy inference system for forecasting of water quality parameters. **Journal of Hydroinformatics**, v. 10, n. 2, p. 127–142, 2008.
- PEREIRA, L. M.; COSTA, R. S. Inteligência artificial para transformação do monitoramento ambiental. **Ciência e Tecnologia Ambiental**, Instituto de Pesquisas Ambientais, v. 12, n. 4, p. 78–95, 2022.
- PIJANOWSKI, B. C.; BROWN, D. G.; SHELLITO, B. A.; MANTHENA, G. A. Using neural networks and gis to forecast land use changes: a land transformation model. **Computers, Environment and Urban Systems**, v. 26, n. 6, p. 553–575, 2002.
- POWELL, M. J. D. Radial basis functions for multivariable interpolation: A review. In: **Algorithms for Approximation**. [S.l.]: Clarendon Press, 1987. p. 143–167.
- RODRIGUES, A. M.; ALMEIDA, S. R. Desafios ambientais contemporâneos e soluções tecnológicas. **Sustentabilidade e Inovação**, Revista Sustentável, v. 18, n. 3, p. 112–129, 2022.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, p. 386–408, 1958.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning internal representations by error propagation. In: **Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations**. [S.l.]: MIT Press, 1986. p. 318–362.
- SANTOS, J. P.; LIMA, A. C. Fundamentos teóricos de redes neurais para aplicações ambientais. **Teoria e Prática em IA Ambiental**, Editora Teoria, v. 6, n. 3, p. 67–85, 2021.
- SILVA, A. B. Monitoramento ambiental e sustentabilidade: desafios e perspectivas. **Revista Brasileira de Gestão Ambiental**, Editora Ambiental, v. 15, n. 3, p. 45–62, 2020.
- SOUZA, C. D.; OLIVEIRA, M. F. Aplicações de machine learning no processamento de dados ambientais. **Revista de Inteligência Artificial e Meio Ambiente**, Sociedade Brasileira de IA, v. 8, n. 2, p. 123–145, 2021.
- SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: A simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, v. 15, p. 1929–1958, 2014.