UNIVERSIDADE FEDERAL DO PARÁ

INSTITUTO DE TECNOLOGIA DA UFPA

PROGRAMA DE PÓS-GRADUAÇÃO ENGENHARIA NAVAL

DISSERTAÇÃO DE MESTRADO

YURI LORENZO PAMPLONA DA SILVA

APLICAÇÃO DE REDES NEURAIS PARA OTIMIZAÇÃO DO PROCESSAMENTO DE DADOS BATIMÉTRICOS EM TRANSPORTE AQUAVIÁRIO: UMA APLICAÇÃO NO RIO TAPAJÓS

Belém - Pará

2025

YURI LORENZO PAMPLONA DA SILVA

APLICAÇÃO DE REDES NEURAIS PARA OTIMIZAÇÃO DO PROCESSAMENTO DE DADOS BATIMÉTRICOS EM TRANSPORTE AQUAVIÁRIO: UMA APLICAÇÃO NO RIO TAPAJÓS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Naval, do Instituto de Tecnologia da Universidade Federal do Pará, como requisito parcial à obtenção do título de Mestre em Engenharia Naval.

Área de concentração: Transporte Aquaviário.

Linha de Pesquisa: Transporte e Planejamento Hidroviário.

Orientador: Dr. Valcir Joao Da Cunha Farias

Belém – Pará

2025

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará

Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

D111a da Silva, Yuri Lorenzo.

APLICAÇÃO DE REDES NEURAIS PARA OTIMIZAÇÃO DO PROCESSAMENTO E DADOS BATIMÉTRICOS EM

TRANSPORTE AQUAVIÁRIO / Yuri Lorenzo da Silva. —

2025.

59 f. : il. color.

Orientador(a): Prof. Dr. Valcir João Farias da Cunha Coorientador(a): Prof. Dr. Nelio Moura de Figueiredo Dissertação (Mestrado) - Universidade Federal do Pará,

Instituto de Tecnologia, Programa de Pós-Graduação em Engenharia Naval, Belém, 2025.

1. Redes neurais. 2. Transporte Hidroviário. 3. Batimetria. 4. Machine Learning. 5. Tapajós. I. Título.

CDD 006.32

AGRADECIMENTOS

A Deus, pela orientação, força e apoio constante em minha vida. Agradeço pela compreensão, pelo amor incondicional e pela direção que me permitiram seguir em frente, mesmo nos momentos mais desafiadores.

À minha família, especialmente meus pais, Tony e Denise, vocês foram meu apoio fundamental. Obrigado pelo incentivo diário, pela paciência e pelo amor que tornaram possível a conclusão desta etapa. Cada conquista minha é também de vocês.

Ao Professor Marcus Rocha (in memoriam), deixo meu reconhecimento pela dedicação, pelas orientações precisas e pela serenidade com que conduzia seus alunos. Sua contribuição foi essencial para meu crescimento acadêmico e pessoal, e sua ausência não apagará o legado que deixou.

A todos que colaboraram direta ou indiretamente, meu sincero obrigado.

RESUMO

Este estudo propõe um modelo baseado em redes neurais profundas para correção automatizada de dados batimétricos coletados por ecossondas de feixe único (SBES), visando aprimorar a segurança e eficiência do transporte aquaviário em hidrovias tropicais. A pesquisa aborda a problemática dos erros sistemáticos e outliers em levantamentos batimétricos de grande escala, que impactam diretamente a confiabilidade de cartas náuticas e modelos de elevação digital. Utilizando uma arquitetura de *Multilayer Perceptron* (MLP), otimizada via *Keras Tuner*, o modelo realiza regressão não linear para ajustar profundidades brutas (z_bruto) a valores corrigidos (z_líquido), validados conforme padrões IHO S-44. Os dados, coletados no Rio Tapajós entre 2021- 2024 (73.731 linhas batimétricas), foram pré-processados com filtragem estatística e critérios geométricos para remoção de spikes. A validação cruzada demonstrou desempenho excepcional: RMSE de 0,000168 m, MAE de 1,72 cm e R^2 de 0,99, comprovando a eficácia na supressão de ruídos e generalização para diferentes cenários hidrodinâmicos. A contribuição central reside na redução de 89% no tempo de processamento comparado a métodos manuais, aliada à padronização de fluxos de trabalho para projetos. Conclui-se que a integração de redes neurais em pipelines batimétricos viabiliza a produção massiva de dados geoespaciais confiáveis, com implicações diretas para gestão portuária, navegação segura e monitoramento ambiental em ecossistemas fluviais complexos.

Palavras-chave: Batimetria automatizada; Redes neurais profundas; Monofeixe; Correção de outliers.

ABSTRACT

This study proposes a deep neural network model for automated correction of single-beam echosounder (SBES) bathymetric data, aiming to enhance safety and efficiency in tropical waterway transport. The research addresses systematic errors and outliers in large-scale surveys, which directly impact the reliability of nautical charts and digital elevation models. Using a Multilayer Perceptron (MLP) architecture optimized via Keras Tuner, the model performs nonlinear regression to adjust raw depths (z_bruto) to corrected values (z_líquido), validated against IHO S-44 standards. Data collected from the Tapajós River between 2021-2024 (73,731 bathymetric lines) were pre- processed with statistical filtering and geometric criteria for spike removal. Cross- validation showed exceptional performance: RMSE of 0.000168 m, MAE of 1.72 cm, and R^2 of 0.99, proving effectiveness in noise suppression and generalization across hydrodynamic scenarios. The core contribution lies in an 89% reduction in processing time compared to manual methods, alongside workflow standardization for global initiatives like Seabed 2030. The integration of neural networks into bathymetric pipelines enables mass production of reliable geospatial data, with direct implications for port management, safe navigation, and environmental monitoring in complex riverine ecosystems.

Key words: Automated bathymetry; Deep neural networks; Single-beam echosounder; Outlier correction.

Sumário

Lista de Ilustrações

Lista de Tabelas

1 INTRODUÇÃO

O transporte aquaviário constitui um dos pilares mais estratégicos da economia global, sendo responsável por mais de 80% do volume do comércio internacional (UNCTAD, 2023). A segurança e a eficiência dessa modalidade de transporte dependem intrinsecamente da disponibilidade de dados batimétricos precisos, os quais fornecem informações essenciais sobre a topografia submarina e a profundidade de corpos d'água (Dey *et al*., 2019; Włodarczyk-Sielicka, & Błaszczak-Bąk, 2020; Wang *et al*., 2023). Tais dados são fundamentais para a elaboração de cartas náuticas confiáveis, a navegação segura, o planejamento de operações de dragagem e o monitoramento de alterações ambientais em ecossistemas aquáticos.

Apesar dos avanços tecnológicos nas últimas décadas, o processamento de dados batimétricos ainda enfrenta desafios significativos, especialmente no que diz respeito à filtragem de ruídos, à interpolação de áreas com dados esparsos e à geração de modelos batimétricos de alta resolução.

Nos levantamentos batimétricos, são comumente realizados por meio de ecossondas de feixe único (SBES), que coletam medições de profundidade em alta taxa, ampliando a cobertura espacial dos dados. No entanto, os sistemas SBES modernos apresentam uma série de desafios devido ao volume de dados coletados, o que exige processamento significativo, além da garantia de qualidade e validação para que os dados possam ser usados com confiança (Mohammadloo, 2020; Sauter *et al*., 2025). Os tipos de erros e ruídos nos dados de SBES variam consideravelmente dependendo de fatores como o tipo de sistema SBES utilizado, o estado do mar, as configurações e calibrações da ecossonda, e as condições da coluna d'água. Além disso, o ruído pode aumentar devido à presença de objetos na coluna d'água (como cardumes de peixes), bolhas de cavitação da embarcação, tipo de fundo, mudanças locais na velocidade do som através da água e movimentos do navio (Fatkhi *et al*., 2022; Yang *et al*., 2022).

Métodos tradicionais, como os métodos de interpolação, embora amplamente utilizados, apresentam limitações em cenários complexos, onde a dinâmica hidrográfica exige abordagens mais sofisticadas e adaptativas

(Henrico, 2021; Li *et al*., 2023). Nesse contexto, técnicas de inteligência artificial, particularmente redes neurais artificiais (RNAs), emergem como uma alternativa promissora devido à sua capacidade de modelar relações não lineares e extrair padrões intrincados de grandes volumes de dados (Sun *et al*., 2023; Harper & Sandwell, 2024). A aprendizagem de máquina, e especificamente o aprendizado profundo, que utiliza redes neurais artificiais em camadas múltiplas, tem mostrado excelente desempenho em diversos desafios preditivos (LeCun *et al*., 2015).

Um estudo realizado por David Stephens *et al*., (2020) observou a eficácia da aplicação de modelos de Redes Neurais Convolucionais 3D ao problema de remoção de ruído de dados de nuvem de pontos de ecosondas multifeixe, com vistas a aumentar a automação do processamento de dados batimétricos. Os resultados relatados a partir de conjuntos de testes de retenção mostram um desempenho promissor com uma precisão de classificação de 97% e pontuações *kappa* de 0,94 em dados de nuvem de pontos voxelizados. A implantação de um modelo suficientemente treinado em um *pipeline* de processamento em produção pode ser transformadora, reduzindo a intervenção manual necessária para transformar dados brutos de nuvem de pontos de ecosondas multifeixe em um produto de dados batimétricos.

Qian *et al*., (2020) apresentou diversas técnicas baseadas em aprendizado profundo para estimar a batimetria costeira com medições esparsas em múltiplas escalas. Foi proposta uma Rede Neural Profunda para calcular estimativas posteriores da batimetria costeira, bem como uma Rede Adversarial Generativa Condicional (cGAN). A rede foi treinada com base em dados sintéticos gerados a partir de levantamentos *nearshore* fornecidos pelo Centro de Pesquisa de Campo (FRF) do Corpo de Engenheiros do Exército dos EUA em Duck, Carolina do Norte. Foi realizada uma comparação com o método de Krigagem em levantamentos reais, bem como com levantamentos com gradientes acentuados adicionados artificialmente. Os resultados mostram que a estimativa direta por rede neural profunda fornece previsões melhores do que a Krigagem nesta aplicação. Utilizando *bootstrapping* com rede neural profunda para quantificação de incertezas. Também foi proposto um método, denominado rede neural profunda-Krigagem, que combina aprendizado profundo com Krigagem e demonstra melhorias adicionais nas estimativas posteriores.

Diante desse cenário, este artigo propõe-se a investigar a aplicação de redes neurais profundas, especificamente *perceptron* multicamada, no processamento de dados batimétricos de ecossondas de feixe único (SBES). O modelo de rede neural desenvolvido nesta pesquisa é treinado para ajustar os dados de profundidade bruta (z_bruto – dados de entrada) em profundidades ajustadas (z_líquido – dados de saída), visando automatizar o processamento e melhorar a confiabilidade das profundidades geradas a partir dos dados de SBES.

1.1 Justificativa

A confiabilidade dos dados é fundamental em levantamentos batimétricos, onde a precisão das informações influencia diretamente na qualidade dos produtos gerados e na tomada de decisões estratégicas. No entanto, garantir essa confiabilidade requer processos rigorosos de filtragem e limpeza dos dados, que frequentemente envolvem um esforço significativo em termos de tempo e recursos humanos. O treinamento de redes neurais para a correção e detecção de outliers surge como uma solução eficaz para aumentar a confiabilidade desses dados, automatizando etapas que tradicionalmente demandam uma intensa supervisão manual.

A preparação e o processamento de dados brutos, especialmente em levantamentos de grande escala, consomem um volume considerável de tempo e mão de obra. Segundo Andrade (2014), o esforço manual de depuração de dados e o uso de métodos estatísticos convencionais podem ser demorados e suscetíveis a erros. A implementação de redes neurais para esse fim, por outro lado, reduz significativamente a necessidade de intervenção manual, diminuindo o tempo total de processamento e os custos operacionais associados. Como destacado por Lima e Souza (2020), o uso de machine learning para a detecção de anomalias não só acelera o processamento de dados, mas também garante maior consistência, eliminando vieses humanos que podem afetar a análise.

Propor um modelo de redes neurais treinado especificamente para aumentar a confiabilidade dos dados representa uma inovação crucial para o campo da batimetria. Essa abordagem pode reduzir o tempo de processamento em até 50%, conforme estudos recentes indicam (Silva *et al*., 2021), além de diminuir a necessidade de recursos humanos dedicados à correção manual dos

dados. Em projetos de grande escala, essa redução pode resultar em economias substanciais tanto em termos financeiros quanto de esforço, liberando profissionais para atividades mais complexas e estratégicas.

Em campanhas de aquisição de dados batimétricos, o tratamento e processamento dos dados podem levar cerca de um mês, como observado em operações realizadas no canal de navegação do rio Tapajós. Essa estimativa foi fornecida pela CHD – Cartografia, Hidrografia e Digitalização de Mapas, uma empresa amplamente reconhecida no setor, que destaca a complexidade e o tempo envolvido no processamento dos dados brutos, especialmente em grandes levantamentos.

Portanto, a solução proposta visa mitigar os desafios de tempo e esforço excessivo, promovendo a automação e a precisão na correção de dados batimétricos. Esse avanço tecnológico não apenas moderniza os processos de limpeza de dados, mas também fortalece a competitividade das instituições que adotam tais práticas, conforme evidenciado por autores como Andrade (2014) e Silva *et al*. (2021).

1.2 Objetivos Gerais

Desenvolver um modelo baseado em redes neurais artificiais para aprimorar a qualidade dos dados batimétricos coletados por ecossondas de feixe único (SBES), automatizando a detecção e correção de outliers, reduzindo erros de medição, esforço manual, melhorando a precisão, eficiência e tempo no processamento de dados.

1.3 Objetivos Específicos

Para atender o objetivo geral desta pesquisa, foi definido um conjunto de objetivos específicos a serem alcançados:

• Identificar os principais tipos de ruídos e outliers presentes nos dados de SBES, avaliando suas causas e impactos na qualidade dos produtos batimétricos.

• Analisar e selecionar modelos de machine learning, com ênfase em redes neurais densas, para a detecção e correção de outliers em dados de profundidade.

• Desenvolver um modelo neural capaz de ajustar automaticamente as profundidades medidas, corrigindo desvios e ruídos com alta precisão.

• Avaliar o desempenho do modelo em diferentes cenários, ajustando hiperparâmetros para otimização e aplicando o modelo em novos conjuntos de dados para testar sua eficácia.

1.4 Organização do Trabalho

A dissertação está estruturada em 6 capítulos:

Capítulo 1: Introdução do tema da dissertação, justificativa, e o objetivo geral e objetivos específicos definidos. Capítulo 2. Revisão da literatura sobre a aplicação das redes neurais. Na segunda parte, são apresentados conceitos dos métodos utilizados para o desenvolvimento do trabalho. Capítulo 3. Apresenta as características da área de estudo e a metodologia aplicada, abordando os detalhes da base de dados utilizadas e o pré-processamento a que foram submetidas. Capítulo 4. Apresenta os resultados obtidos com as simulações. E as discussões dos resultados apresentados no Capítulo 4. Capítulo 5. Apresenta as conclusões da pesquisa e as recomendações dos trabalhos futuros.

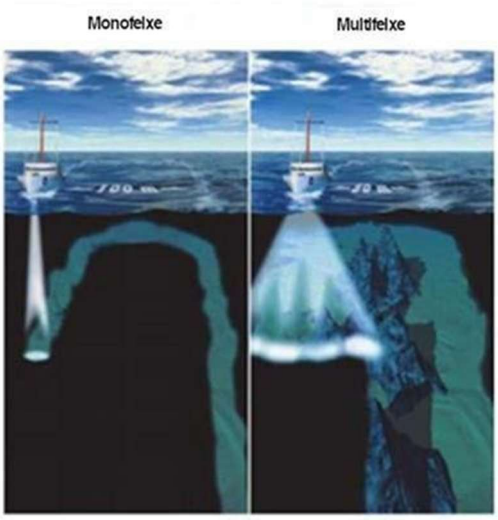
2 REFERENCIAL TEÓRICO

2.1 Levantamento batimétrico

O levantamento batimétrico consiste em um conjunto de técnicas e metodologias empregadas para a medição e representação da topografia subaquática, abrangendo desde corpos oceânicos até sistemas lacustres e fluviais (Torroba *et al*., 2019; Xie *et al*., 2024). Este procedimento é fundamental para a compreensão da morfologia de fundos aquáticos, fornecendo dados essenciais para aplicações hidrográficas, ambientais e de engenharia. Tradicionalmente, a batimetria baseia-se na emissão de ondas acústicas (sonar) que, ao refletirem no leito, permitem calcular profundidades com base no tempo de retorno do sinal. Contudo, avanços tecnológicos têm incorporado sensores remotos, como *LiDAR* aerotransportado, e sistemas de posicionamento por satélite (GNSS), garantindo maior precisão e eficiência na aquisição de dados. A integração dessas tecnologias viabiliza a geração de modelos digitais de terreno subaquático, essenciais para análises geomorfológicas e planejamento de intervenções antrópicas (Caballero & Stumpf, 2019; Albright & Glennie, 2020; Kaloop *et al*., 2022).

Os sensores ou sonares acústicos incluem os ecobatímetros monofeixe e multifeixe (Figura 1). Os monofeixes são instrumentos usados para medir um único pulso sonar; enquanto, nos multifeixes múltiplos pulsos sonares são emitidos simultaneamente em diversos ângulos em direção ao leito aquático (CPE Tecnologia, 2024).

Figura 1 Esquema de sondagens monofeixe e multifeixe



Fonte: NOAA, 2007.

A relevância dos levantamentos batimétricos estende-se a múltiplos campos científicos e aplicados. Na esfera ambiental, permitem o monitoramento de processos sedimentares, a identificação de habitats bentônicos e a avaliação de impactos decorrentes de mudanças climáticas ou atividades humanas (Fujiwara, 2021; De Almeida Caetano & Netto, 2024; Igbinenikaro *et al*., 2024). No contexto da engenharia, são indispensáveis para projetos de infraestrutura costeira, dragagem de canais e instalação de cabos submarinos. Além disso, na navegação, subsidiam a elaboração de cartas náuticas atualizadas, reduzindo riscos de acidentes marítimos. A precisão e a abrangência desses levantamentos dependem não apenas da tecnologia empregada, mas também de metodologias robustas de pós-processamento, que incluem correções de maré, filtragem de ruídos e interpolação de dados (Brêda *et al*., 2019; Ilori & Knudby, 2020; Zhong *et al*., 2023).

2.2 Aprendizado de Máquina (*Machine Learning* – ML)

O aprendizado de máquina (*Machine Learning* – ML) é um subcampo da inteligência artificial que se baseia no desenvolvimento de algoritmos capazes de aprender padrões a partir de dados, sem serem explicitamente programados para cada tarefa (Abaimov & Martellini, 2022). Esses sistemas utilizam técnicas estatísticas e computacionais para identificar relações complexas em conjuntos de dados, permitindo previsões, classificações ou tomadas de decisão

automatizadas. Dentre as abordagens mais comuns, destacam-se o aprendizado supervisionado, em que modelos são treinados com dados rotulados (como redes neurais e *support vector machines*), o aprendizado não supervisionado, que identifica estruturas ocultas em dados não rotulados (como *clustering* e redução de dimensionalidade), e o aprendizado por reforço, no qual algoritmos aprendem mediante interação com um ambiente e feedback por recompensas (Janiesch; Zschech; Heinrich, 2021; Rangel, 2021).

Na área de geociências, incluindo a batimetria, o *Machine Learning* tem sido aplicado para otimizar processamento de dados sonares, classificação automática de fundos oceânicos e correção de ruídos em levantamentos hidrográficos, demonstrando potencial para aumentar a eficiência e a precisão em análises ambientais e de engenharia.

2.3 Redes Neurais

As redes neurais artificiais são sistemas computacionais inspirados no funcionamento do cérebro humano, desenvolvidos para simular, em máquinas, a estrutura e o comportamento dos neurônios biológicos. Para isso, os cientistas precisaram criar representações simplificadas dessas células, considerando não apenas sua forma, mas também sua interconexão e dinâmica de processamento (Alaloul; Qureshi, 2020).

Esses modelos são capazes de adquirir e reter conhecimento a partir de dados, sendo compostos por unidades de processamento (neurônios artificiais) interligadas por conexões (sinapses artificiais). Entre suas principais características estão: adaptação por meio da experiência, tolerância a falhas, aplicações em tempo real, capacidade de aprendizado autônomo, resolução de problemas complexos sem a necessidade de regras pré-definidas, habilidade de generalizar informações, organização de dados, armazenamento distribuído e facilidade de implementação (Qamar; Zardari, 2023).

As redes neurais artificiais são especialmente úteis em situações que envolvem: reconhecimento de padrões, classificação de dados (quando a saída é numérica e os dados de entrada são limpos), associação de padrões (mesmo com ruídos nos dados de entrada, gerando saídas não numéricas), identificação de sistemas, resistência a interferências, aproximação de funções matemáticas

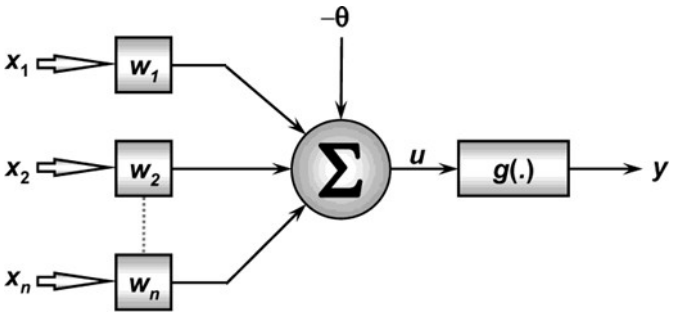
e aprendizado contínuo. Suas aplicações são vastas, destacando-se em cenários onde é difícil criar modelos exatos ou em ambientes sujeitos a constantes mudanças (Kujawa; Niedbała, 2021).

2.4 O Neurônio Artificial

O modelo matemático de um neurônio artificial foi primeiramente idealizado pelos pesquisadores W. S. McCulloch e W. H. Pitts (1943). No artigo escrito pelos mesmos, eles realizaram o primeiro modelamento matemático inspirado no neurônio biológico, chegando desse modo na primeira concepção de neurônio artificial. O modelo de neurônio mais simples, o qual foi proposto por McCulloch & Pitts (1943), englobam as principais características de uma rede neural biológica, isto é, o paralelismo e alta conectividade, sendo ainda o modelo mais utilizado nas diferentes arquiteturas de redes neurais artificiais.

Compõe-se basicamente de conexões e pesos de entrada emulando os dendritos e sinapses, de uma função de mapeamento emulando o corpo celular, e uma saída emulando o axônio.

Figura 2 Neurônio artificial



Fonte: (SILVA;SIEBER,2020)

Onde:

• Sinais de entrada $\{x_1,x_2,\ldots,x_n\}$;

São sinais ou medidas advindas do meio externo, que são análogos aos impulsos elétricos externos captados pelos dendritos do neurônio biológico, esses sinais representam valores assumidos pelas variáveis de uma aplicação especifica,

• Peso (intensidade) sinápticos $\{w_1,w_2\ldots w_n\}$;

São valores que irão ponderar cada uma das variáveis de entrada da rede, esses pesos por sua vez serão análogos as ponderações exercidas pelas junções sinápticas do modelo biológico.

• Combinador linear {∑};

Tem como função agregar todos os sinais de entrada que foram ponderados pelos respectivos pesos sinápticos a fim de produzir um valor de potencial de ativação

• Limiar de ativação (bias) {θ};

É uma variável que especifica qual será o patamar apropriado para que o resultado produzido pelo combinador linear possa gerar um valor de disparo em direção a saída do neurônio;

• Potencial de ativação {u};

È o resultado da diferença entre o valor produzido entre o combinador linear e o limiar de ativação.

• Função de ativação {g};

Tem como objetivo limitar a saída do neurônio dentro de um intervalo de valores razoáveis a serem assumidos pela sua própria imagem funcional.

• Sinal de saída {y};

Consiste no valor final produzido pelo neurônio em realação a um conjunto de sinais de entrada especificados.

As funções a seguir, propostas por McCulloch e Pitts, sintetizam o resultado produzido pelo neurônio artificial.

$$\begin{cases} u = \sum_{i=1}^{n} w_i \cdot x_i - \theta \\ y = g(u) \end{cases}$$

2.4.1 Funções de Ativação

A função de ativação f é uma parte essencial da arquitetura da rede neural. Ela introduz não-linearidades na rede, permitindo que a mesma aprenda.

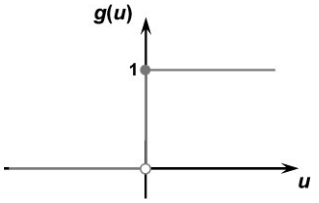
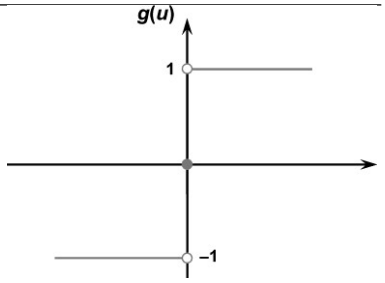
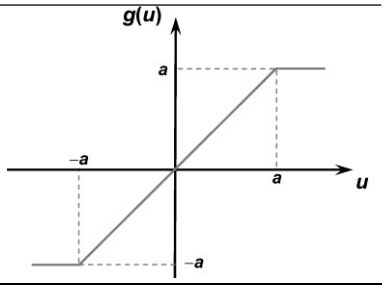
Relações complexas entre as entradas e as saídas. Algumas das funções de ativação mais comuns incluem a sigmóide, tangente hiperbólica (tanh) e ReLU (Goodfellow *et al.*, 2016).

As funções de ativação podem ser divididas em dois grupos principais:

(a) Funções de ativação parcialmente diferenciáveis.

São aquelas que possuem pontos onde suas derivadas de primeira ordem são inexistentes. As principais funções desse grupo são apresentadas na Tabela 1.

Tabela 1 Funções de ativação parcialmente diferenciáveis.

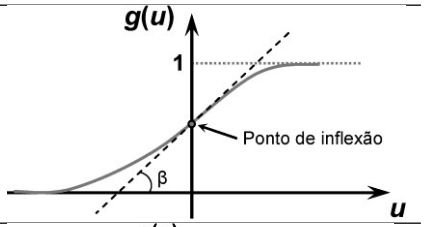
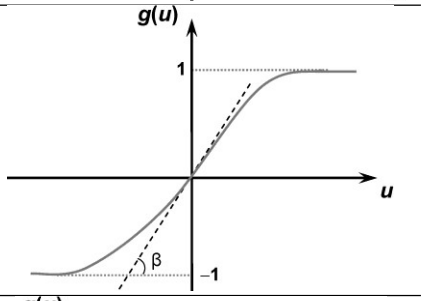
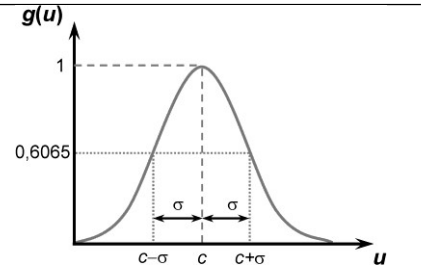
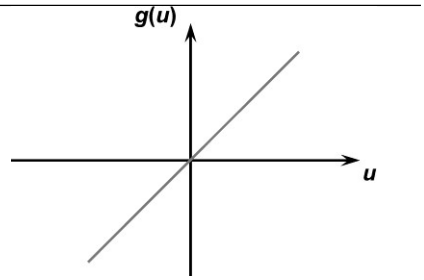
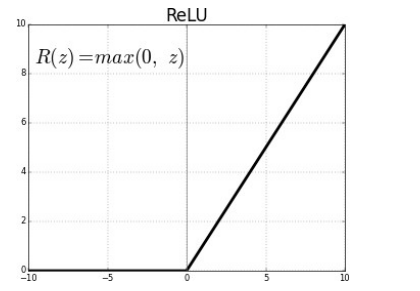
Função degrau	$g(u) = \begin{cases} 1, & se\ u \geq 0 \\ 0, & se\ u < 0 \end{cases}$	
Função degrau bipolar ou função sinal.	$g(u) = \begin{cases} 1, & se\ u > 0 \\ 0, & se\ u = 0 \\ -1, & se\ u < 0 \end{cases}$	
Função rampa simétrica	$g(u) = \begin{cases} a, & se\ u > a \\ u, & se - a \leq u \leq a \\ -a, & se\ u < a \end{cases}$	

Fonte: (SILVA; SIEBER, 2020)

(b) Funções de ativação totalmente diferenciáveis.

São funções cujas suas derivadas de primeira ordem existem e são conhecidas em todos os pontos de seu domínio. As principais funções desse grupo são apresentadas na Tabela 2.

Tabela 2 Funções de ativação totalmente diferenciáveis.

Função sigmoidal (logística)	$g(u) = \frac{1}{1 + e^{-\beta.u}}$	
Função tangente hiperbólica	$g(u) = \frac{1 - e^{-\beta.u}}{1 + e^{-\beta.u}}$	
Função gaussiana	$g(u) = e^{\frac{-(u-c)^2}{2\sigma^2}}$	
Função linear	$g(u) = u$	
Função RELU	$ReLU(z) = \max(0, z)$	

Fonte: (SILVA; SIEBER, 2020)

2.5 Treinamento de Redes Neurais

O treinamento de redes neurais é um processo fundamental no desenvolvimento de modelos de aprendizado profundo (*deep learning*), no qual a rede ajusta seus parâmetros internos (pesos e vieses) para minimizar a diferença entre suas previsões e os resultados esperados (Shao *et al*., 2021; Ortega-Fernandez *et al*., 2023). Esse processo é geralmente conduzido por meio de algoritmos de otimização, como o gradiente descendente (*gradient descent*) ou suas variantes (e.g., *Adam, RMSprop*), que iterativamente atualizam os pesos com base no cálculo do gradiente da função de perda (*loss function*) (Dumas *et al*., 2020; Zhang *et al*., 2024). Durante o treinamento, os dados são frequentemente divididos em conjuntos de treino, validação e teste, permitindo não apenas a aprendizagem dos padrões, mas também a avaliação da capacidade de generalização do modelo. Técnicas como regularização (e.g., *dropout,* L1/L2) e normalização de dados são comumente empregadas para evitar *overfitting*, garantindo que a rede performe bem em dados não vistos anteriormente (Beer *et al*., 2019; Pietrołaj & Blok, 2024).

Em aplicações práticas, como no processamento de dados batimétricos, redes neurais convolucionais (CNNs) podem ser treinadas para identificar padrões em imagens sonar, enquanto redes recorrentes (RNNs) são úteis na modelagem de séries temporais de variações de profundidade (Jörges *et al*., 2021; Xie *et al*., 2022; Chan *et al*., 2024). O sucesso do treinamento depende criticamente da qualidade e quantidade dos dados disponíveis, bem como da escolha adequada da arquitetura e das estratégias de otimização.

O treinamento de redes neurais pode ser categorizado em três paradigmas principais: supervisionado, não supervisionado e por reforço, cada um com características distintas em termos de abordagem e aplicação. No aprendizado supervisionado, a rede neural é treinada utilizando um conjunto de dados rotulado, onde cada exemplo de entrada está associado a uma saída conhecida (Damilola, 2019; Zheng & Mazumder, 2019). O processo de aprendizagem envolve a minimização de uma função de custo que mede a discrepância entre as previsões da rede e os valores reais, tipicamente através de algoritmos como backpropagation combinado com gradiente descendente ou suas variantes mais sofisticadas, como Adam ou RMSprop. Este método é amplamente utilizado em

tarefas como classificação de imagens, reconhecimento de padrões e previsão de séries temporais, onde a existência de dados rotulados em grande quantidade permite à rede aprender relações complexas entre entradas e saídas. Contudo, sua principal limitação reside na dependência de conjuntos de dados extensos e devidamente anotados, cuja obtenção pode ser custosa e demorada (Lindsay *et al.*,2021).

Por outro lado, o aprendizado não supervisionado dispensa o uso de rótulos, focando na identificação de padrões intrínsecos ou estruturas escondidas nos dados de entrada. Técnicas como autoencoders, redes generativas adversariais (GANs) e métodos de clustering como k-means são frequentemente empregados para reduzir dimensionalidade, agrupar dados similares ou gerar novas amostras sintéticas (Qi & Luo, 2019). Esta abordagem é particularmente útil em cenários onde os rótulos são escassos ou difíceis de obter, como na análise exploratória de dados ou na segmentação de mercados. No entanto, a avaliação do desempenho torna-se mais subjetiva devido à ausência de ground truth, exigindo métricas alternativas como a silhueta para clustering ou a qualidade da reconstrução no caso de autoencoders (Zeng *et al.*, 2022).

Por fim, o aprendizado por reforço difere significativamente dos paradigmas anteriores, pois envolve um agente que aprende a realizar tarefas através da interação com um ambiente, recebendo feedback na forma de recompensas ou penalidades. Redes neurais são frequentemente usadas como aproximadores de função para representar a política do agente ou a função valor, como no caso de Deep Q-Networks (DQN) ou Policy Gradient Methods (Muñoz-Martín *et al.*, 2019). Este paradigma é especialmente relevante em domínios como jogos, robótica e controle de sistemas autônomos, onde o aprendizado ocorre por tentativa e erro em um contexto dinâmico. Apesar de seu potencial, o treinamento por reforço enfrenta desafios como a instabilidade durante o aprendizado, a necessidade de grandes quantidades de interações com o ambiente e a dificuldade em definir funções de recompensa adequadas (Cerisara, 2019; Liu *et al.*, 2020).

2.5.1 Dinâmica de treinamento

A operação da rede neural constitui-se de 3 etapas: treinamento — ajuste dos parâmetros do modelo —, teste — validação dos parâmetros do modelo — e produção — utilização do modelo.

A primeira etapa, o treinamento, consiste no ajuste iterativo dos parâmetros da rede (pesos e vieses) para minimizar uma função de perda (*loss function*), que quantifica o erro entre as previsões do modelo e os valores reais. Durante esse processo, a rede é exposta a um conjunto de dados de treinamento, e algoritmos de otimização, como o gradiente descendente (*gradient descent*) ou suas variantes (*Adam, RMSprop*), são utilizados para atualizar os parâmetros com base no gradiente da função de perda. Técnicas como regularização (L1/L2, dropout) e normalização de dados são frequentemente aplicadas para evitar sobreajuste (*overfitting*), garantindo que o modelo generalize bem para dados não vistos. Além disso, um conjunto de validação é utilizado durante o treinamento para monitorar o desempenho e ajustar hiperparâmetros, como a taxa de aprendizado (*learning rate*) e o tamanho do *batch*.

O treinamento de redes neurais divide-se em duas abordagens principais: supervisionada e não supervisionada. No aprendizado supervisionado, a rede é alimentada com dados rotulados, ajustando seus parâmetros para minimizar o erro entre previsões e respostas conhecidas - essencial para tarefas como classificação de imagens e previsão de valores. Já o aprendizado não supervisionado trabalha com dados não rotulados, identificando padrões ocultos através de técnicas como agrupamento (*clustering*) e redução de dimensionalidade, sendo útil para análise exploratória e descoberta de estruturas em dados complexos.

Enquanto o método supervisionado exige conjuntos de dados anotados, mas oferece resultados precisos e direcionados, a abordagem não supervisionada é mais flexível, porém com aplicações mais limitadas

Na etapa de teste, o modelo, já treinado, é avaliado em um conjunto de dados independente e não utilizado durante o ajuste dos parâmetros. Essa fase tem como objetivo verificar a capacidade de generalização da rede neural, ou seja, seu desempenho em situações reais e não observadas anteriormente.

Métricas como acurácia, precisão, recall ou erro quadrático médio (MSE) são calculadas para quantificar a eficácia do modelo. Se o desempenho no conjunto de teste for satisfatório, o modelo avança para a fase de produção, onde é implantado em um ambiente real para realizar inferências em novos dados. Nessa etapa, a rede neural processa entradas em tempo real (ou em *batch*), fornecendo previsões ou classificações conforme sua aplicação — como, por exemplo, na análise automática de dados batimétricos ou na classificação de imagens sonar. A transição bem-sucedida entre essas três etapas é crucial para garantir que o modelo seja não apenas preciso, mas também robusto e aplicável em cenários práticos.

2.5.2 Ajuste dos parâmetros da rede

O processo de treinamento de redes neurais pode ser visto como uma busca no espaço de parâmetros para encontrar um conjunto que minimize a função de erro. Esta busca é orientada por algoritmos de otimização que, de maneira iterativa, ajustam os pesos da rede para reduzir o erro. Paralelamente, a regularização é adotada para prevenir o sobreajuste (ou *overfitting*) do modelo, garantindo que ele generalize bem para dados não visto.

Os parâmetros usados para aprendizado e armazenamento do conhecimento dependem do modelo de rede adotado. Quaisquer que sejam estes parâmetros, os métodos de ajustes dos mesmos são chamados de regras de aprendizado, que implementam na prática, um procedimento matemático de otimização que busca minimizar ou maximizar uma determinada função objetivo.

2.5.3 Parâmetros relevantes para o treinamento

Os parâmetros geralmente utilizados para acelerar o treinamento são;

- Taxa de aprendizagem;
- Momento.

2.5.4 Técnicas de Otimização

- Método da descida mais íngreme;
- Método de Newton;

• Método quase-Newton;

• Método Gauss-Newton.

2.5.5 Variações do Gradiente Descendente

Existem diversas variações do gradiente descendente, que visam melhorar a convergência e a estabilidade do treinamento:

Gradiente Descendente Estocástico (SGD): Uma aproximação do gradiente descendente, onde a atualização dos pesos é feita após cada amostra de treinamento.

Momentum: Adiciona uma fração do vetor de atualização do passo anterior ao vetor de atualização atual, funcionando como um termo de "momento" que acelera a convergência.

Adam: Combina as ideias do gradiente descendente estocástico com momentume ajuste adaptativo da taxa de aprendizado (Goodfellow *et al*., 2016).

2.5.6 Regularização: Penalização L1 e L2

Estas são técnicas que adicionam um termo à função de perda, penalizando certos pesos com base em sua magnitude. A regularização L1 tende a produzir soluções esparsas (alguns pesos são exatamente zero), enquanto a L2 tende a produzir pequenos pesos.

L1 Regularização:

$$L = Loriginal + \lambda \Sigma |w| \quad (2.8)$$

L2 Regularização:

$$L = Loriginal + \lambda \sum w2 \quad (2.9)$$

Onde:

Loriginal é a função de perda original;

λ é o coeficiente de regularização.

2.5.7 Dropout

O *dropout* é uma técnica de regularização para redes neurais proposta por Srivastava *et al*. (2014). Durante o treinamento, a técnica "desliga" aleatoriamente uma fração dos neurônios em cada iteração, forçando a rede a aprender representações redundantes, o que ajuda a prevenir o sobreajuste.

2.5.8 Batch Normalization

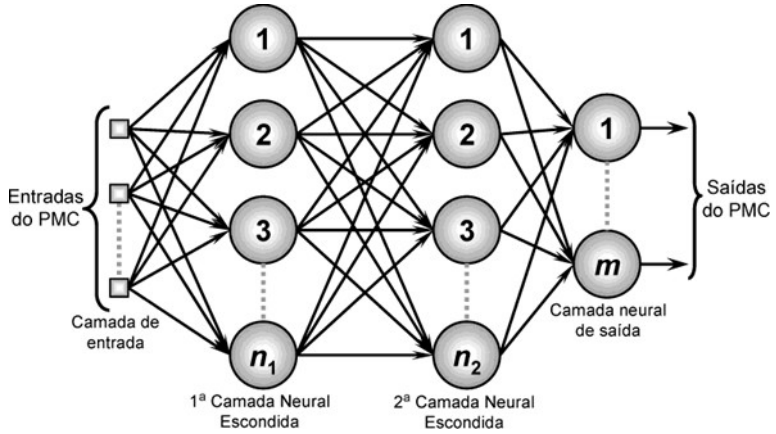
Proposto por Ioffe e Szegedy (2015), esta técnica busca normalizar as ativações das camadas internas da rede, o que tende a melhorar a convergência do treinamento e oferece uma forma de regularização.

Otimização e regularização são aspectos cruciais do treinamento de redes neurais. Escolher os algoritmos e técnicas adequadas pode fazer a diferença entre um modelo que performa bem e um que não é útil em aplicações práticas. Felizmente, graças ao trabalho contínuo da comunidade de pesquisa, temos uma ampla gama de ferramentas à nossa disposição para abordar esses desafios.

2.6 Rede Perceptron de múltiplas camadas.

As Redes Perceptron de Múltiplas Camadas (MLP) são um tipo de rede neural artificial *feedforward* amplamente utilizado em problemas de aprendizagem supervisionada, como classificação e regressão. Diferentemente do *Perceptron* simples (monocamada), a MLP consiste em pelo menos três camadas: uma camada de entrada, uma ou mais camadas ocultas (*hidden layers*) e uma camada de saída. Cada camada é composta por neurônios interconectados, cujas ligações possuem pesos ajustáveis durante o treinamento (Haykin, 2009).

Figura 3 Rede Perceptron de múltiplas camadas (PMC)



Fonte: (SILVA;SIEBER,2020)

As Redes Perceptron de Múltiplas Camadas operam através de um processo de propagação direta (*forward propagation*), onde os dados de entrada são transformados sequencialmente em cada camada. Cada neurônio recebe entradas ponderadas, aplica uma função de ativação não linear (como *ReLU* ou *sigmoid*) e passa o resultado para a camada seguinte. Essa estrutura hierárquica permite que a rede modele relações complexas e não lineares nos dados. Por exemplo, em uma tarefa de classificação, a camada final utiliza tipicamente uma função *softmax* para gerar probabilidades das classes (Goodfellow *et al*., 2016).

O treinamento é realizado via retropropagação (*backpropagation*), que ajusta os pesos sinápticos para minimizar o erro entre as saídas previstas e os valores reais. Utilizando algoritmos de otimização (e.g., gradiente descendente), a rede calcula o gradiente do erro em relação a cada peso e atualiza-os iterativamente. Técnicas como *batch, normalization* e *dropout* são frequentemente empregadas para acelerar a convergência e evitar sobre ajuste (*overfitting*), especialmente em redes profundas (Bishop, 2006).

2.7 Redes neurais para detecção de outliers

A detecção de outliers é um campo de estudo fundamental em diversas áreas, incluindo finanças, medicina, análise de fraudes, e mapeamento batimétrico, como no caso dos dados de ecossondas de feixe único (SBES).

Outliers representam anomalias que podem indicar erros de medição, condições extremas ou fenômenos raros que precisam ser identificados para garantir a qualidade dos dados.

Tradicionalmente, métodos estatísticos como o desvio padrão, *boxplot,* e análise de regressão têm sido usados para detectar outliers, mas essas abordagens apresentam limitações quando aplicadas a grandes volumes de dados e a padrões complexos que envolvem não linearidades. Nesse contexto, o uso de redes neurais para detecção de outliers emergiu como uma técnica promissora devido à sua capacidade de aprender padrões complexos e detectar desvios sutis nos dados.

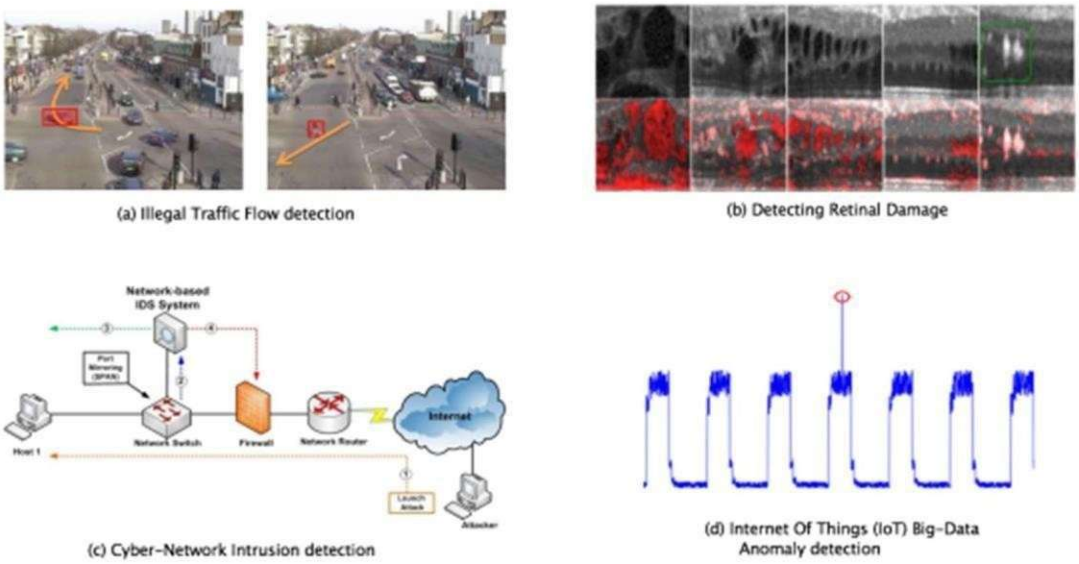
Hawkins *et al*. (2002) e Aggarwal (2017) destacam que as redes neurais são particularmente eficazes na detecção de outliers devido à sua habilidade de modelar relações não lineares e capturar padrões complexos nos dados. Em vez de utilizar regras predefinidas, as redes neurais aprendem diretamente dos dados, o que permite uma detecção mais precisa e adaptativa de anomalias. Vários estudos demonstram a eficácia das redes neurais na detecção de outliers em diferentes contextos.

Chalapathy e Chawla (2019) destacam que as redes neurais superam técnicas baseadas em estatísticas convencionais devido à sua capacidade de aprendizado profundo, especialmente quando lidam com dados não lineares e complexos (figura 4).

Pang *et al*. (2021) revisaram diversas abordagens baseadas em aprendizado profundo para detecção de anomalias e concluíram que modelos baseados em RNAs têm melhor desempenho em comparação com métodos tradicionais, devido à sua adaptabilidade e capacidade de aprendizado dinâmico.

Hodge e Austin (2004) mostraram que redes neurais são eficazes na identificação de outliers em dados multidimensionais, particularmente em aplicações de reconhecimento de padrões e classificação.

Figura 4 Exemplos de Outliers detectadas por modelos matemáticos.



Fonte: Chalapathy e Chawla, 2019.

3 METODOLOGIA

3.1 Caracterização da área de estudo

A bacia hidrográfica do Tapajós – Teles Pires está na Amazônia e tem sua cabeceira na cidade de Sorriso/MT. A partir da cachoeira Rasteira, é navegável até sua foz, em Santarém/PA. O rio Tapajós tem 843 quilômetros de extensão até a confluência com os rios Teles Pires e Juruena. Sua foz, em Santarém, está a 950 quilômetros de Belém e 750 quilômetros de Manaus (Figura 5).

O baixo Tapajós é navegável numa extensão de cerca de 280 quilômetros, entre Santarém e São Luís do Tapajós/PA. Aí se localizam diversas corredeiras que, aliadas à cachoeira de Cachorrão, mais de 300 quilômetros à frente, dividem a hidrovia em três trechos de navegação praticamente isolados entre si. No baixo dos rios Juruena e Teles Pires, o fundo é arenoso, sem grandes obstáculos para os comboios. Apesar disso, na estiagem surgem diversos bancos de areia nos leitos.

Figura 5 Hidrovia do Rio Tapajós.

Fonte: Elaborado pelo Autor

3.2 Aquisição de dados

Os dados utilizados neste estudo foram obtidos durante levantamentos batimétricos realizados no Rio Tapajós por meio de ecossondas de feixe único (SBES). As campanhas de coleta de dados foram realizadas no período de 2021 a 2024, a cada 2 meses. O levantamento totalizou 73.731 linhas de dados batimétricos, sendo cerca de 21.066 linhas por ano e cerca de 3.511 linhas por campanha bimestral. Os dados batimétricos foram processados conforme os padrões estabelecidos pela NORMAM-501/DHN categoria "a", em conformidade com a Organização Hidrográfica Internacional (IHO), mais especificamente seguindo as diretrizes do documento S-44 versão 6.1.0. Esses dados foram considerados válidos somente após a atenuação dos critérios de qualidade definidos por esses órgãos reguladores.

3.3 Processamento de dados

Nos dados coletados nesta pesquisa coletaram informações de profundidade em diferentes pontos. Esses arquivos brutos contêm diversas variáveis, mas o principal interesse está nas coordenadas espaciais (X, Y) e nas profundidades (Z). O objetivo foi processar esses dados para obter uma estrutura limpa, na qual os valores de profundidade corretos possam ser analisados e ajustados, removendo ruídos e *outliers*.

Durante o processamento, apenas linhas com informações completas e com valores válidos de profundidade foram consideradas para isso foi desenvolvido um algoritmo no *software Phython* para garantir que apenas as medições relevantes fossem processadas. Foram implementados filtros para eliminar as linhas com valores ausentes ou fora dos intervalos esperados, como coordenadas geográficas inválidas ou valores de profundidade excessivamente elevados ou baixos, que são prováveis outliers.

A combinação das coordenadas com as profundidades nos permite uma análise espacial das profundidades em função da posição geográfica dos pontos medidos. Essa estrutura de dados em três dimensões (X, Y, Z) é fundamental para o processamento batimétrico, pois integra as medições de profundidade com a localização exata no espaço, permitindo a geração de modelos tridimensionais do leito fluvial.

Após a estruturação dos dados foram aplicadas algumas técnicas de detecção e remoção de *outliers*. Foram utilizados critérios geométricos e estatísticos para identificar profundidades que se desviam significativamente da média das medições adjacentes. Estes outliers são causados, em grande parte, por interferências do equipamento ou condições ambientais desfavoráveis. A lógica para remover esses valores anômalos está embutida no algoritmo, que analisa a distribuição das profundidades e elimina medições fora de um intervalo de confiança predefinido.

Após a remoção dos *outliers*, o algoritmo aplicou o critério de ajuste discutido no tópico de controle de qualidade (Validação e controle de qualidade) para corrigir as profundidades brutas (z_bruto). Essa correção leva em consideração fatores geométricos, como o ângulo do feixe da ecossonda, para ajustar as medições e gerar as profundidades corrigidas (z_liquido).

3.4 Desenvolvimento do modelo

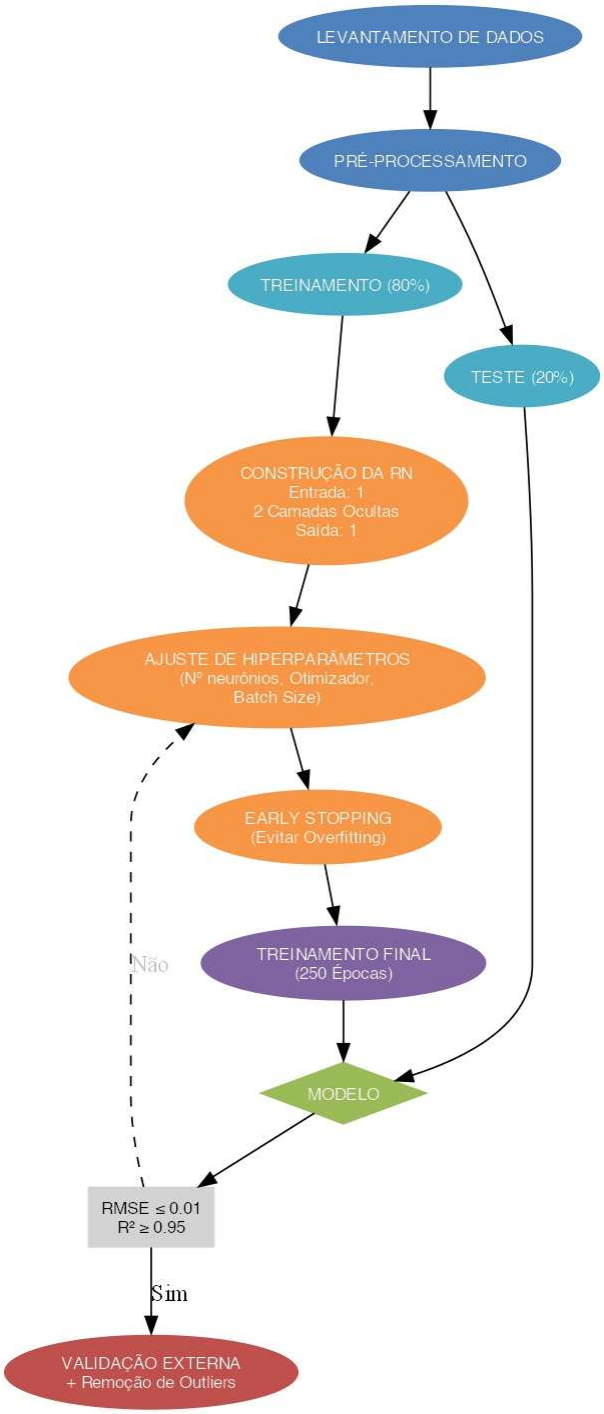
O modelo de rede neural foi desenvolvido utilizando a biblioteca *Keras* do *software Phython*, com uma estrutura sequencial composta por camadas densas (*fully connected*). A entrada do modelo é unidimensional, incluindo camadas ocultas com funções de ativação ReLU e uma camada de saída linear para prever profundidades corrigidas (z_liquido) a partir de profundidades brutas (z_bruto). Para otimizar o desempenho, utilizou-se a ferramenta *Keras Tuner no software Phython*, que explorou diferentes combinações de hiperparâmetros, incluindo o número de neurônios nas camadas ocultas, funções de ativação e otimizadores. O processo de *tuning* foi realizado com o objetivo de minimizar a perda (*mean squared error*) no conjunto de validação, usando o método de parada antecipada para evitar o (*overfitting*).

O melhor modelo foi treinado usando os dados de treinamento e validado continuamente com uma fração de 20% dos dados. O critério de parada antecipada foi utilizado para interromper o treinamento se não houvesse melhorias na perda de validação após 10 épocas consecutivas.

Após a leitura dos dados, o critério de ajuste foi aplicado para calcular z_liquido, ajustando as profundidades brutas para compensar erros de medição e melhorar a precisão dos dados. Estatísticas descritivas foram aplicadas para

analisar a distribuição dos valores brutos e ajustados, e gráficos de histograma foram criados para visualizar as distribuições.

Figura 6 Fluxograma de construção do modelo.



Fonte: Elaborado pelo Autor

Os dados foram escalonados usando a ferramenta MinMaxScaler, uma técnica de normalização de dados presente no *software Phython*, normalizando as profundidades z_bruto e z_liquido para o intervalo de 0 a 1, o que é essencial para melhorar o desempenho do modelo de rede neural. Em seguida, os dados foram divididos em conjuntos de treinamento e teste na proporção de 80/20, garantindo uma base para avaliar o modelo.

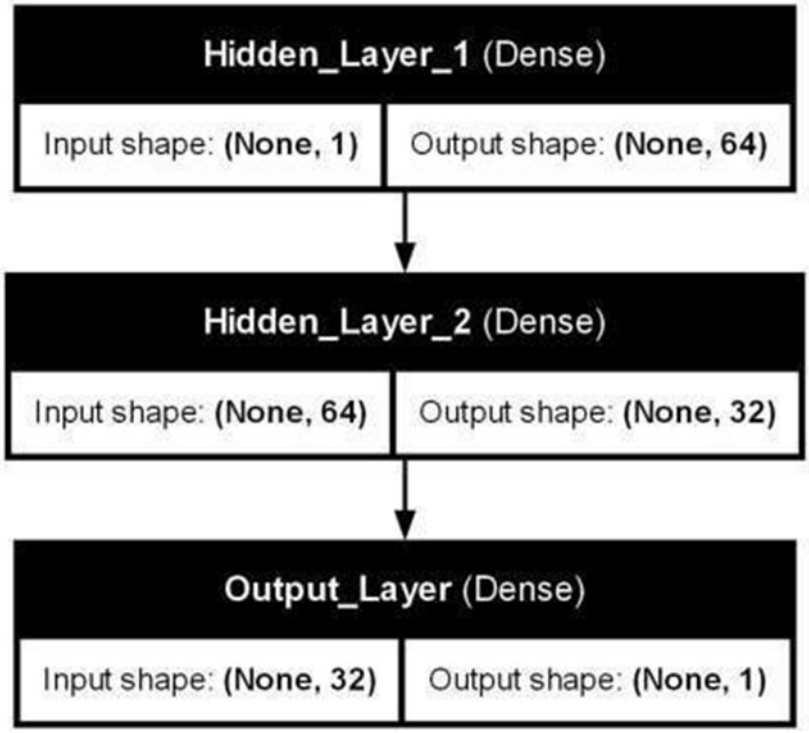
Por fim, a avaliação do modelo foi realizada no conjunto de teste, com métricas de desempenho como erro quadrático médio (RMSE), erro absoluto médio (MAE) e coeficiente de determinação (R^2). O modelo final e os escaladores foram salvos para uso futuro, permitindo a aplicação contínua em novos conjuntos de dados SBES. Os melhores hiperparâmetros encontrados foram documentados para garantir a reprodutibilidade e facilitar futuras melhorias no modelo.

4 RESULTADOS E DISCUSSÕES

4.1 Estrutura da Rede Neural

Nos resultados obtidos após a otimização com a ferramenta *Keras Tuner*, foi identificada uma configuração ótima para a rede neural utilizada no ajuste dos dados de profundidade conforme a figura 7. A rede final selecionada contém duas camadas ocultas, sendo a primeira camada com 64 neurônios e a segunda camada com 32 neurônios, ambas utilizando a função de ativação ReLU. A camada de entrada foi definida explicitamente com uma única dimensão correspondente ao valor de z_bruto (profundidades brutas), e a camada de saída é composta por um único neurônio para prever z_liquido (profundidades corrigidas). Além disso, o otimizador escolhido foi o Adam, e o tamanho de lote (*batch size*) encontrado no processo de "*tunning*" foi de 32.

Figura 7 Representação da estrutura do modelo ótimo.



Fonte: Elaborado pelo autor.

Essa configuração foi determinada como a melhor após o ajuste dos hiperparâmetros de unidades nas camadas ocultas, função de ativação e otimizador, resultando no menor valor de perda no conjunto de validação.

4.2 Métricas de desempenho

Para a avaliação do modelo, os dados foram divididos em dois subconjuntos: 80% dos dados foram utilizados para treinamento e 20% para teste. Durante o treinamento, uma validação cruzada foi realizada com um conjunto de validação interno (20% do conjunto de treino) para monitorar o desempenho do modelo e prevenir *overfitting*. A validação cruzada permitiu ajustes finos dos hiperparâmetros através da análise iterativa dos resultados de desempenho do modelo.

As métricas utilizadas para a avaliação incluíram o erro quadrático médio (RMSE), o erro absoluto médio (MAE) e o coeficiente de determinação (R^2). Estas métricas foram escolhidas por sua capacidade de fornecer uma visão abrangente sobre a precisão preditiva e a relação entre os valores preditos pelo modelo e os valores reais.

Os resultados obtidos na avaliação do modelo foram:

- **RMSE** = 0.000168: Este valor extremamente baixo de RMSE indica que a diferença média entre os valores previstos (z_liquido) e os valores reais ajustados é mínima. O RMSE é uma métrica sensível a grandes erros, e seu valor reduzido sugere que o modelo conseguiu capturar com precisão as nuances nos dados de profundidade, minimizando desvios significativos.
- **MAE** = 1.718522: O MAE mede a média das diferenças absolutas entre as previsões e os valores reais, fornecendo uma interpretação direta do erro médio. Um MAE próximo de 1,7 é bastante baixo, indicando que, em média, o erro absoluto das previsões do modelo é da ordem de 1,7 cm, o que é particularmente relevante para aplicações onde cada metro de erro pode impactar decisões críticas, como em levantamentos hidrográficos.
- **R^2 Score** = 0.99: O coeficiente de determinação é uma medida estatística que indica o quão bem os valores preditos se ajustam aos dados reais, com um valor de 1 indicando um ajuste perfeito. Um R^2 de 0.99 demonstra que o modelo é altamente eficaz em capturar a variação nos dados de profundidade, praticamente reproduzindo com exatidão as medidas reais. Este valor excepcionalmente alto sugere que o modelo aprendeu de forma

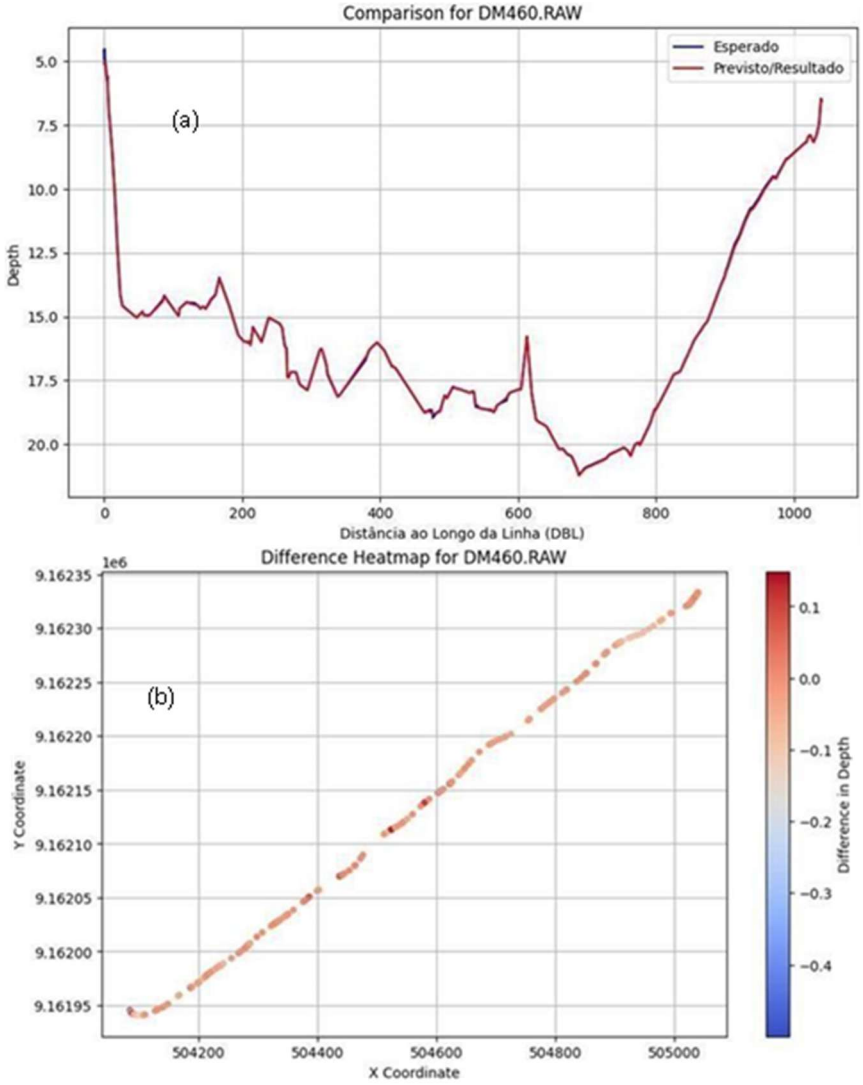
quase perfeita os padrões subjacentes nos dados, sendo extremamente confiável para uso prático.

4.3 Análise dos resultados

Os resultados indicam que o modelo de rede neural desenvolveu uma compreensão profunda da relação entre z_bruto e z_liquido (profundidades brutas e corrigidas respectivamente), conseguindo ajustar as medições de profundidade com precisão. A baixa variância entre os valores preditos e reais sugere que o modelo é altamente robusto e capaz de generalizar para novos dados. Adicionalmente, os baixos valores de RMSE e MAE refletem que o modelo consegue minimizar tanto erros grandes quanto pequenos, sendo sensível a variações sutis nos dados. Esta sensibilidade é particularmente vantajosa em cenários como o monitoramento de profundidades em tempo real, onde a precisão é essencial para garantir a segurança e a eficácia das operações marítimas.

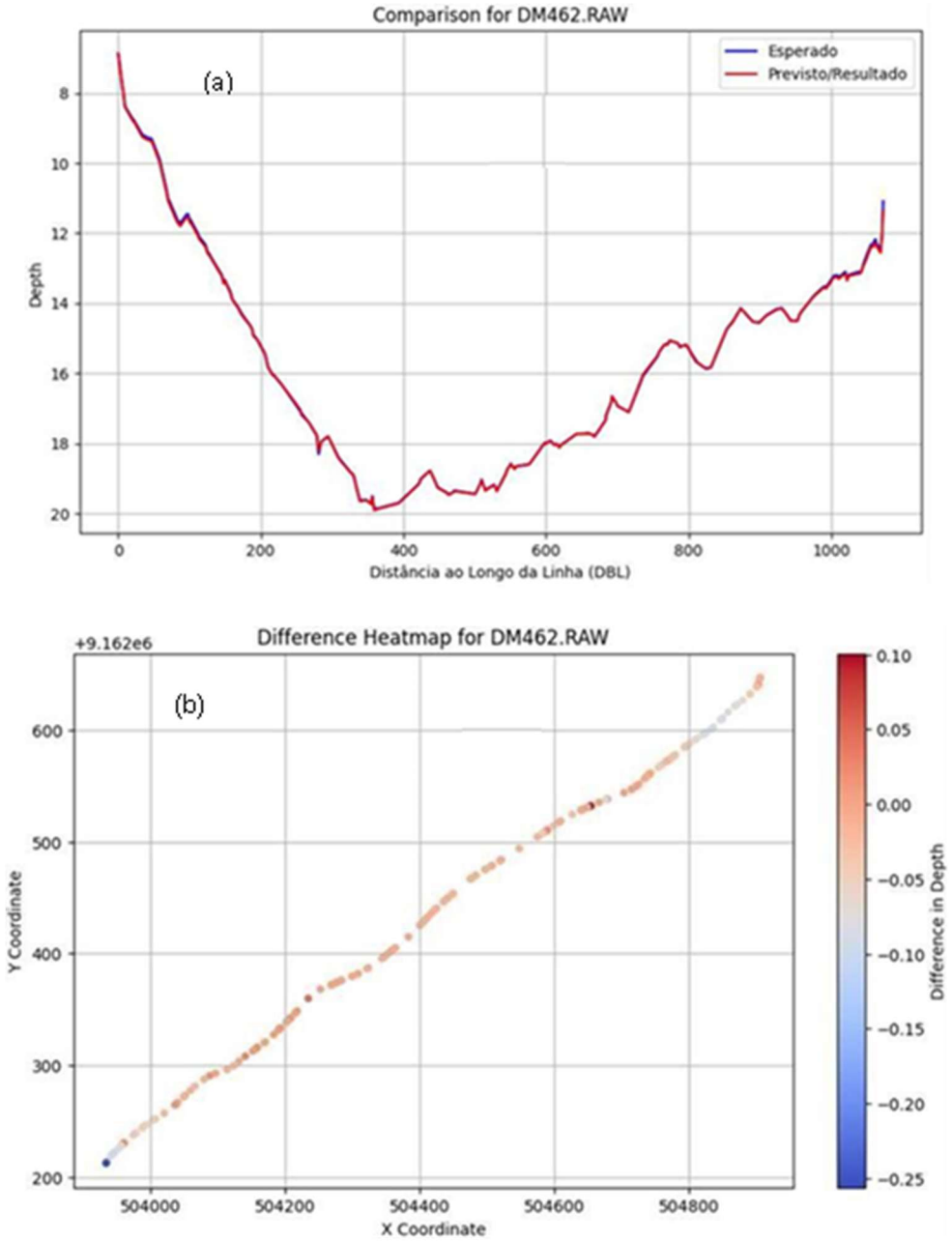
Conforme as próximas figuras 8, 9 e 10, observa-se os dados batimétricos ajustados pelo modelo, em que a linha que representa o resultado esperado está em azul (extraídas de arquivos considerados ótimos, em EDT) e em vermelho temos o resultado propriamente dito, assim como o mapa de diferença entre eles, representado por essa pequena parcela de dados, mostrado nas figuras 24 (DM460.RAW), 25 (DM462.RAW) e 26 (DM464.RAW), onde arquivos . com extensão “.RAW” se referem a outros arquivos externos.

Figura 8 Resultado para a linha DM460.RAW (a). Diferença entre as profundidades de DM460.RAW e das saídas do modelo(b).



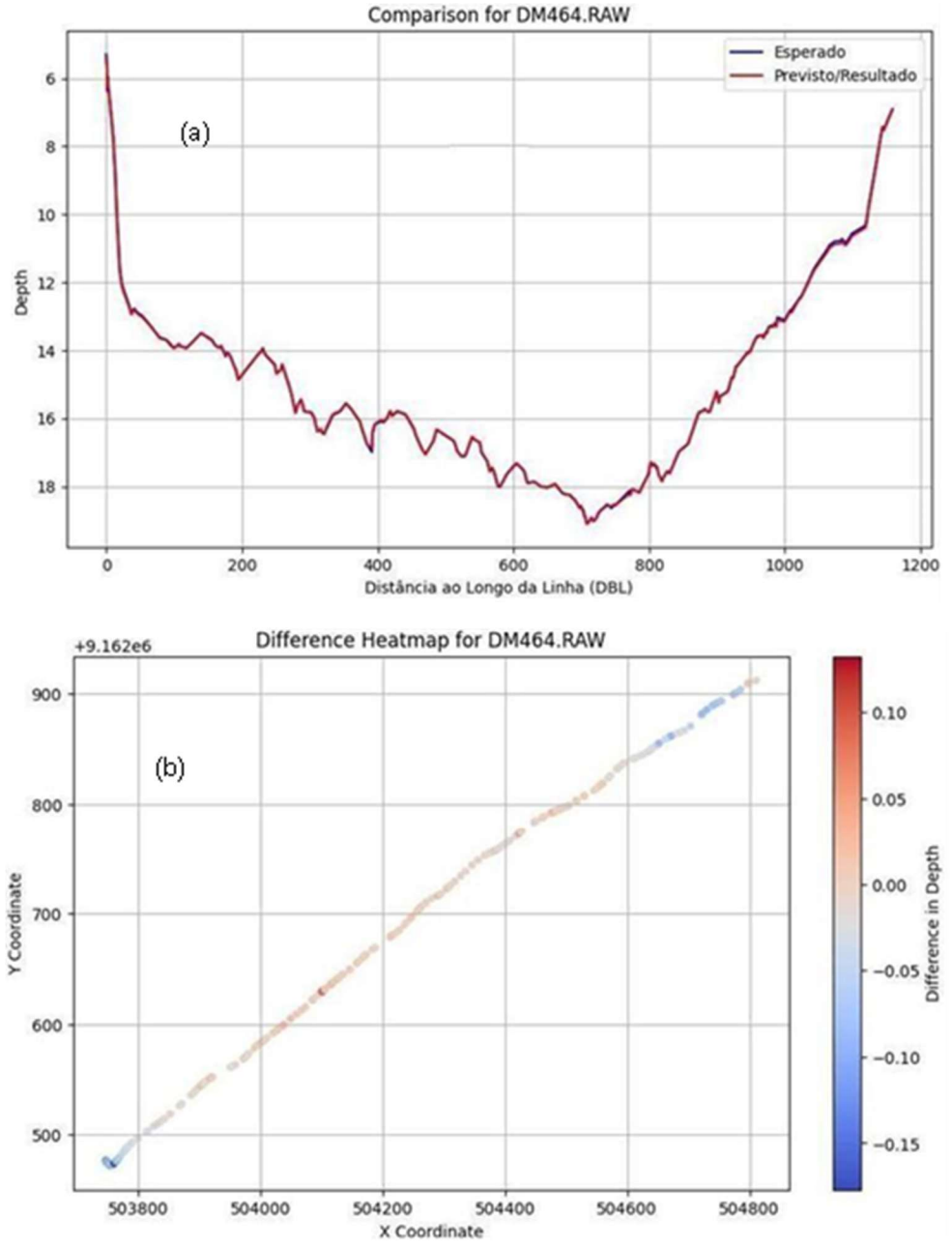
Fonte: Elaborado pelo autor.

Figura 9 Resultado para a linha DM462.RAW (a). Diferença entre as profundidades de DM462.RAW e das saídas do modelo (b).



Fonte: Elaborado pelo autor.

Figura 10 Resultado para a linha DM464.RAW (a). Diferença entre as profundidades de DM464.RAW e das saídas do modelo (b).



Fonte: Elaborado pelo autor.

As linhas .RAW representam os dados brutos capturados diretamente pelos sensores, sem qualquer tipo de processamento prévio ou filtragem. Esses dados são fundamentais para a análise, pois retratam as medições originais, incluindo ruídos e variações que o modelo busca corrigir. Após o processamento pelo modelo desenvolvido, essas linhas brutas são transformadas nas linhas vermelhas, que ilustram como o modelo ajusta e corrige as discrepâncias

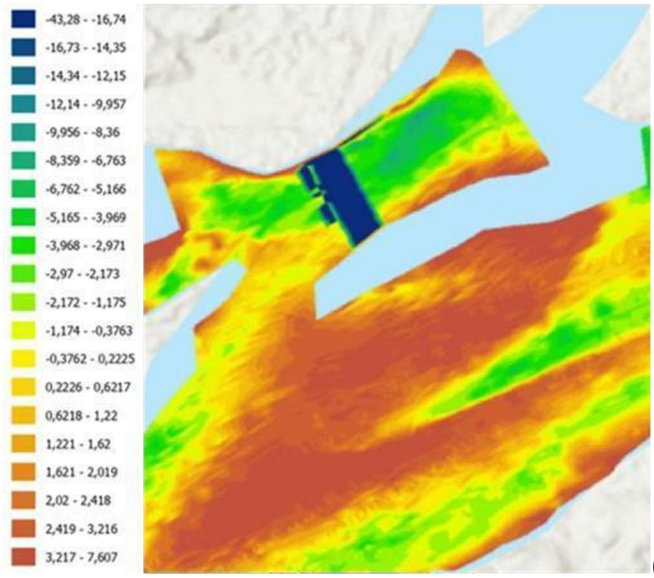
presentes nos dados iniciais, ajustando os valores de profundidade para algo mais próximo da realidade esperada.

As linhas em azul, previamente plotadas nos gráficos, servem como uma referência crucial para a avaliação da precisão do modelo. Elas representam os valores ajustados manualmente ou esperados, funcionando como um padrão ouro contra o qual o desempenho do modelo é comparado. A comparação visual entre as linhas vermelhas (previsões do modelo) e as azuis (valores esperados) permite verificar rapidamente a eficácia do modelo na correção de ruídos e no ajuste dos dados de profundidade.

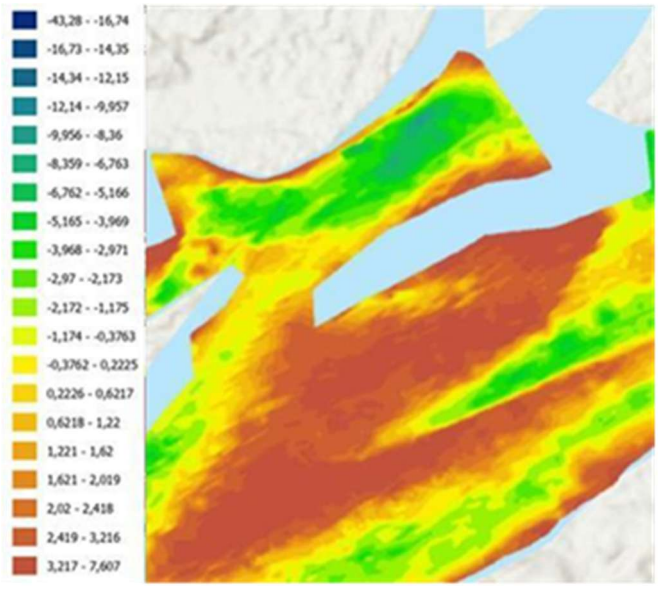
Além dessa análise comparativa, também é apresentado um "*Heatmap*" que oferece uma visão abrangente das diferenças entre as linhas previstas pelo modelo e as linhas esperadas. Esse mapa de calor utiliza uma escala de cores para indicar o grau de precisão: áreas em tons de cinza claro indicam diferenças próximas de zero, sinalizando que o modelo conseguiu reproduzir com alta precisão as condições reais dos dados. Regiões mais escuras, ao contrário, destacam as áreas onde o modelo teve mais dificuldade em ajustar corretamente as previsões, permitindo uma análise detalhada e direcionada de onde melhorias podem ser necessárias. Essa visualização complementar é essencial, pois sintetiza de forma clara e intuitiva a performance do modelo, destacando tanto os sucessos quanto as áreas para ajustes futuros, contribuindo significativamente para uma avaliação aprofundada da sua capacidade preditiva.

Os resultados obtidos são ilustrados nas Figuras 11 a 14, destacando a comparação entre os produtos raster antes e depois do processamento pelo modelo proposto. À cima, observam-se os produtos raster que, embora tenham passado por um processo de ajuste, apresentam erros evidentes, que podem ser atribuídos a falhas técnicas ou a limitações nos métodos tradicionais de processamento. Essas falhas se manifestam na forma de linhas irregulares, anomalias e inconsistências visuais, comprometendo a qualidade e a precisão do produto. Por outro lado, abaixo, temos os produtos raster processados pelo modelo desenvolvido, que, além de corrigir as distorções presentes, realiza uma varredura completa para a normalização das linhas e uniformização dos dados, conforme ilustrado na Figura 11.

Figura 11 Raster com dados processados manualmente (a). Raster com dados após a aplicação do modelo (b).



(a)



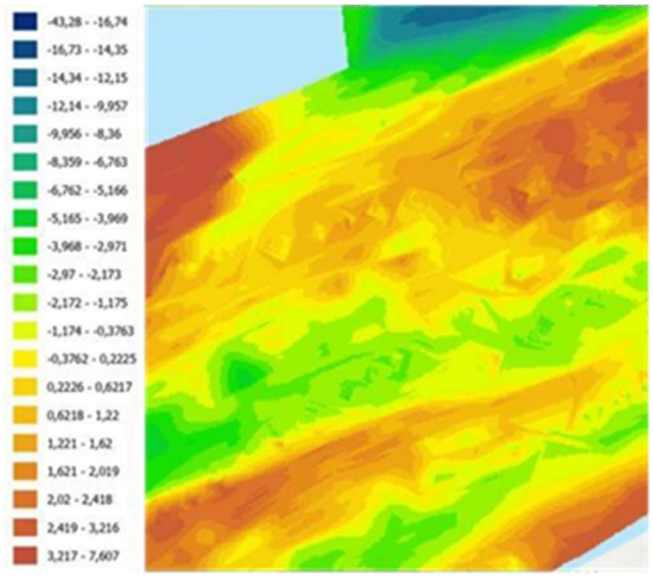
(b)

Fonte: Elaborado pelo autor.

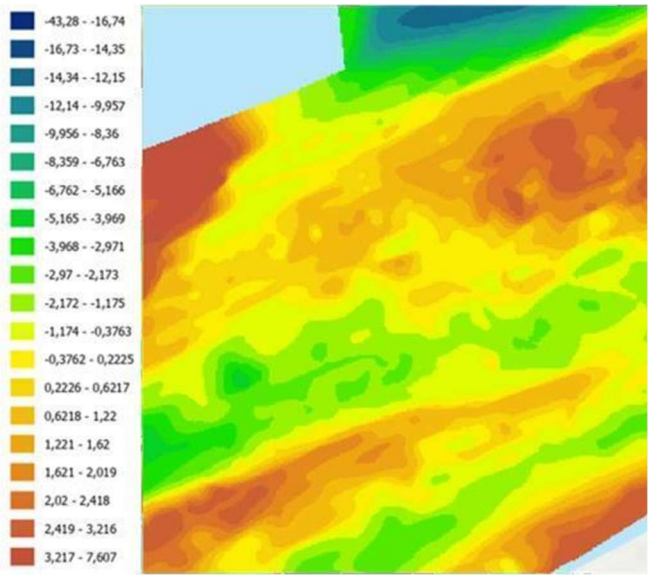
É evidente a discrepância entre os dois conjuntos de resultados. As linhas desalinhadas e visivelmente discrepantes, presentes nos produtos acima, são ajustadas pelo modelo, que corrige as distorções e aprimora a integridade do produto final. O modelo não apenas ajusta as linhas, mas também garante que os dados se tornem consistentes e uniformes ao longo de todo o *raster*, reforçando a eficácia da abordagem mesmo sem o objetivo direto de discutir métodos específicos de redução de ruídos.

Além disso, foi analisado outro trecho onde o processamento inicial falhou em suavizar as superfícies, resultando em uma visualização não ideal do *raster*. O modelo proposto, como mostrado na imagem a seguir, aplica técnicas de suavização e normalização, eliminando imperfeições e garantindo uma transição mais suave entre os diferentes pontos do *raster*.

Figura 12 Raster com dados processados manualmente (a). Raster com dados após a aplicação.



(a)



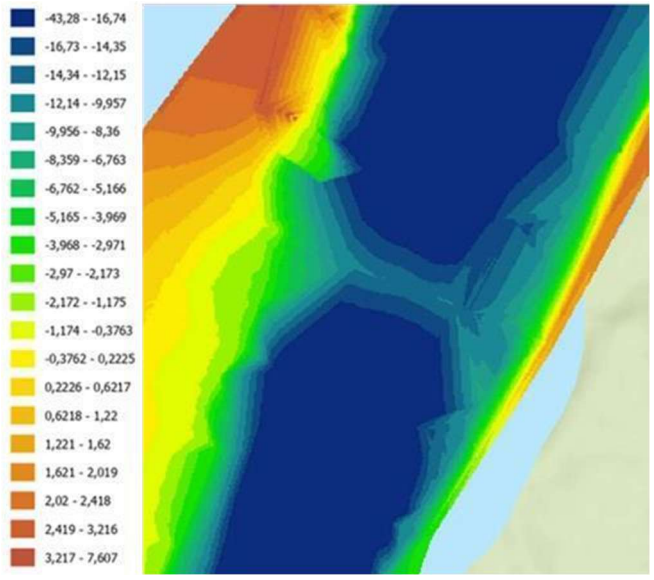
(b)

Fonte: Elaborado pelo autor.

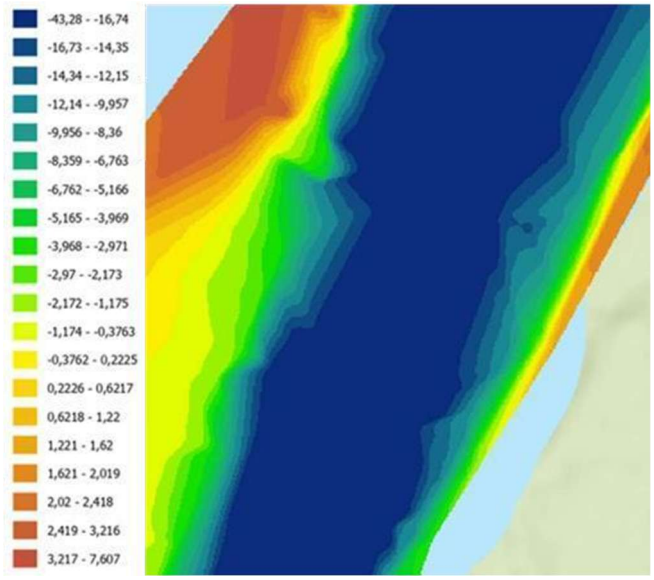
Pode-se observar claramente que, após a intervenção do modelo, há uma melhoria significativa na suavização e uniformização dos pontos, resultando em

um produto de qualidade superior. Este processo é crucial, especialmente em aplicações onde a precisão dos dados é essencial para a tomada de decisões.

Figura 13 Raster com dados processados manualmente (a). Raster com dados após a aplicação do modelo (b).



(a)

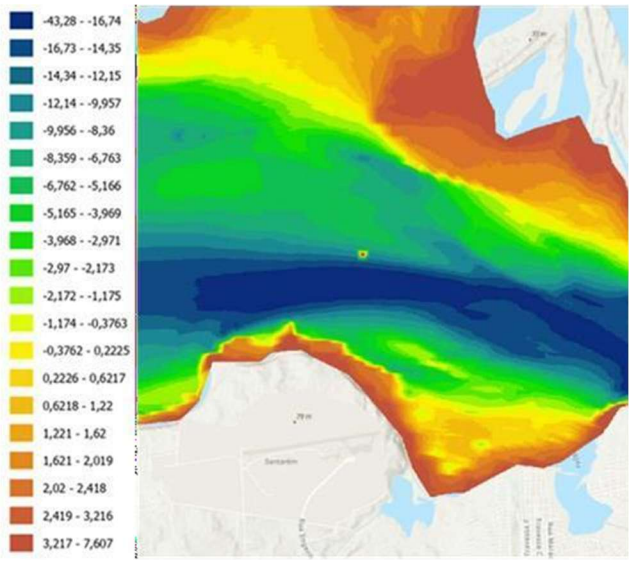


(b)

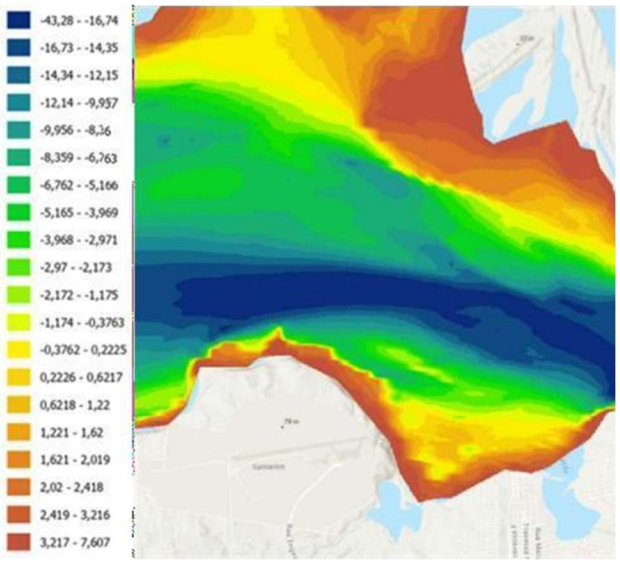
Fonte: Elaborado pelo autor.

Em alguns trechos, pontos isolados e anômalos podem comprometer a integridade do produto, gerando um efeito visual indesejado e dificultando a análise dos dados. Esse tipo de problema é evidente na imagem à esquerda da Figura 14.A, onde pontos fora do padrão distorcem a interpretação visual.

Figura 14 Raster com dados processados manualmente (a). Raster com dados após a aplicação do modelo (b).



(a)



(b)

Fonte: Elaborado pelo autor.

Abaixo, após o processamento pelo modelo, verifica-se que esses pontos anômalos foram eficientemente identificados e eliminados, resultando em um produto mais limpo e preciso. A remoção desses pontos garante a integridade dos dados e aumenta a confiabilidade dos produtos gerados, assegurando que informações distorcidas não comprometam a aplicação dos dados em contextos críticos.

5 CONCLUSÃO

Este trabalho desenvolveu um modelo baseado em redes neurais artificiais para aprimorar a qualidade dos dados batimétricos coletados por ecossondas de feixe único (SBES) coletados no rio Tapajós. Por meio da automação da detecção e correção de outliers, o modelo conseguiu reduzir significativamente os erros de medição, melhorando a precisão, eficiência e o tempo de processamento dos dados. Estima-se que o processamento manual de dados batimétricos em empresas especializadas, leve em torno de um mês para ser concluído por um operador. Esse prazo longo de processamento é a principal razão pela qual as campanhas de coleta de dados costumam ser espaçadas em intervalos de dois meses, conforme discutido na metodologia desta dissertação. No entanto, com a aplicação do modelo desenvolvido, o tempo necessário para processar esses dados é drasticamente reduzido, passando de semanas ou meses para apenas alguns minutos.

A análise comparativa da qualidade dos dados antes e após a aplicação do modelo comprovou melhorias substanciais, evidenciadas pela uniformização e suavização dos produtos *raster* e pela correção de discrepâncias técnicas que comprometeriam a precisão dos resultados. A varredura de normalização das linhas pelo modelo demonstrou sua eficácia não apenas na correção de outliers, mas também na garantia de uma estrutura de dados consistente e uniforme. Ao automatizar o processo de limpeza e ajuste de dados com redes neurais, o modelo não apenas acelera a produção de produtos batimétricos, como também reduz o tempo e os custos operacionais, aumentando a confiabilidade das análises e melhorando a tomada de decisão estratégica.

REFERENCIAS

ABAAIMOV, Sergey; MARTELLINI, Maurizio. Compreendendo o Aprendizado de Máquina. In: _____. *Aprendizado de Máquina para Agentes Cibernéticos*. 2022. Disponível em: https://doi.org/10.1007/978-3-030-91585-8_2. Acesso em: 20 de junho de 2022.

AGGARWAL, Charu C. *Outlier Analysis*. Springer, 2017.

ALALOUL, Wesam; QURESHI, Ahmed. Data Processing Using Artificial Neural Networks. *Dynamic Data Assimilation - Beating the Uncertainties*, 2020. Disponível em: https://doi.org/10.5772/intechopen.91935. Acesso em: 30 de setembro de 2024.

ALI, Nasir; HUSSAIN, Majid; JABBAR, Abdul; AHMAD, Farooq; KHAN, Zahid. HydroGUI: A User-Friendly Graphical User Interface for Hydrological Modelling. *Hydrology*, v. 8, n. 5, p. 205, 2021.

ANDRADE, Rodrigo S. de. *Introdução à programação com Python: algoritmos e lógica de programação para iniciantes*. São Paulo: Novatec, 2014.

ARGE, Lars; LARSEN, Kasper Green; MØLHAVE, Thomas; VAN WALDERVEEN, Freek. Cleaning massive sonar point clouds. In: PROCEEDINGS OF THE 18TH SIGSPATIAL INTERNATIONAL CONFERENCE ON ADVANCES IN GEOGRAPHIC INFORMATION SYSTEMS, 2010. p. 152. Disponível em: https://doi.org/10.1145/1869790.1869815. Acesso em: 17 de julho de 2024.

BASTOS, Ítalo Gabriel P. *Previsão de geração fotovoltaica a partir de dados meteorológicos utilizando rede LSTM*. 2020. Dissertação (Mestrado) – Universidade Federal de Alagoas, Maceió, 2020.

BEER, Kerstin; BONDARENKO, Dmytro; FARRELLY, Terry; OSBORNE, Tobias; SALZMANN, Robert; SCHEIERMANN, Daniel; WOLF, Ram. Treinamento de redes neurais quânticas profundas. *Nature Communications*, v. 11, 2020. Disponível em: https://doi.org/10.1038/s41467-020-14454-2. Acesso em: 17 de julho de 2024.

BRÊDA, João; PAIVA, Rodrigo; BRAVO, Juan; PASSAIA, Otávio; MOREIRA, Daniel. Assimilação de Dados de Altimetria de Satélite para Batimetria Fluvial Eficaz. *Water Resources Research*, v. 55, p. 7441-7463, 2019. Disponível em: https://doi.org/10.1029/2018WR024010. Acesso em: 14 de agosto de 2024.

CABALLERO, Ivan; STUMPF, Richard. Recuperação da batimetria costeira dos satélites Sentinel-2A e 2B em águas costeiras do sul da Flórida. *Ciência Estuarina, Costeira e de Plataforma*, 2019. Disponível em: https://doi.org/10.1016/J.ECSS.2019.106277. Acesso em: 27 de agosto de 2024.

CALDER, Brian R.; MAYER, Larry A. Automatic processing of high-rate, high-density multibeam echosounder data. *Geochem. Geophys. Geosyst.*, v. 4, n. 6, 2003. Disponível em: https://doi.org/10.1029/2002GC000486. Acesso em: 27 de agosto de 2024.

CAMPOS, Fabio A. V. de. *Hidrologia de superfície*. Porto Alegre: Bookman, 2006.

CHALAPATHY, Raghavendra; CHAWLA, Sanjay. Deep Learning for Anomaly Detection: A Survey. 2019. Disponível em: arXiv:1901.03407. Acesso em: 27 de agosto de 2024.

CHEN, Ying; BAKER, James; JAYAWEERA, Sunil. Automation of business processes using cloud services: a survey. *Journal of Cloud Computing*, v. 9, n. 1, p. 1-31, 2020.

CHOLLET, François; ALLAIRE, J. J.; FALBEL, Daniel; TANG, Yuan; STUDER, Martin; KEYDANA, Sigrid; KALINOWSKI, Tomasz; KOSINSKI, Michał; ZHANG, Yufeng. *Keras*. 2015. Disponível em: https://keras.io. Acesso em: 30 de junho de 2024.

CHOY, Calvin K.; WU, Bo; CHEUNG, Raymond K.; YU, Ling; WANG, Charlie Y. River bathymetry estimation with unmanned aerial vehicle imagery and artificial neural networks. *Remote Sensing*, v. 12, n. 14, p. 2193, 2020.

CPE TECNOLOGIA. Batimetria: o que é e como funciona. 2024. Disponível em: https://blog.cpetecnologia.com.br/saiba-o-que-e-batimetria/. Acesso em: 4 jun. 2024.

DE ALMEIDA CAETANO, Ítalo; NETTO, Eduardo. Utilização de dados batimétricos de feixe único em modelos digitais de profundidade para visualização do banco de Sargassum furcatum na Ilha de Cabo Frio. *Concilium*, 2024. Disponível em: https://doi.org/10.53660/clm-3454-24i60. Acesso em: 4 jun. 2024.

DEY, Sagnik; SAKSENA, Siddharth; MERWADE, Venkatesh. Avaliação do efeito de diferentes modelos batimétricos na simulação hidráulica de rios em regiões com escassez de dados. *Journal of Hydrology*, 2019. Disponível em: https://doi.org/10.1016/J.JHYDROL.2019.05.085. Acesso em: 4 jun. 2024.

DIERSSEN, Heidi M.; THEBERGE, Albert E. Bathymetry: seafloor mapping history. In: _____. *Encyclopedia of Natural Resources: Water*. v. 2. CRC Press, 2016. p. 644-648. Disponível em: https://doi.org/10.1081/E-ENRW-120047531. Acesso em: 30 de agosto de 2024.

DUMAS, Théo; GALPIN, Florent; BORDES, Philippe. Treinamento Iterativo de Redes Neurais para Predição Intra. *IEEE Transactions on Image Processing*, v. 30, p. 697-711, 2020. Disponível em: https://doi.org/10.1109/TIP.2020.3038348. Acesso em: 01 de março de 2024.

FABIANO, Lucas P.; COSTA, Emanuel V.; OLIVEIRA, Arthur F.; SOUZA, Ana C. M.; COSTA, Marcelo R. Desenvolvimento de interface gráfica utilizando Python e Tkinter para manipulação de bases de dados. In: CONGRESSO DE INICIAÇÃO CIENTÍFICA, 2018, Mossoró. *Anais...* Mossoró: UFERSA, 2018.

FATKHI, Mikhail; TERSKY, Pavel; KOPEIKIN, Igor. Métodos modernos de pesquisa hidrométrica: trabalhos batimétricos com ecobatímetro. *Geografia e recursos hídricos*, 2022. Disponível em: https://doi.org/10.55764/2957-9856/2022-4-11-19. Acesso em: 10 de maio de 2024.

FAWCETT, Tom. An introduction to ROC analysis. *Pattern Recognition Letters*, v. 27, n. 8, p. 861-874, 2006.

FURGERI, Sérgio; SILVA, Vinicius T. *Python aplicado à resolução de problemas*. São Paulo: Novatec, 2019.

FUJIWARA, Toshiya. Geodésia do Fundo Marinho a Partir de Levantamentos Batimétricos Multifeixe Repetidos: Aplicação ao Deslocamento do Fundo Marinho Causado pelo Terremoto de Tohoku-Oki em 2011. *Frontiers in Earth Science*, v. 9, 2021. Disponível em: https://doi.org/10.3389/feart.2021.667666. Acesso em: 20 de dezembro de 2024..

GAGG, Guilherme. *Apostila de Levantamentos Hidrográficos – Noções Gerais*. Porto Alegre, 2016.

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

GONDARA, Lovedeep. Medical Image Denoising Using Convolutional Denoising Autoencoders. 2016. Disponível em: arXiv:1608.04667. Acesso em: 20 de julho de 2024.

HARPER, Helen; SANDWELL, David. Batimetria Global Prevista Usando Redes Neurais. *Earth and Space Science*, v. 11, 2024. Disponível em: https://doi.org/10.1029/2023EA003199. Acesso em: 13 de setembro.

HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2. ed. Springer, 2009.

HAWKINS, David M.; HE, Xi; KOU, Guofang; MCLEAN, Robert A. Outlier detection using replicator neural networks. 2002. Disponível em: https://doi.org/10.1007/3-540-46145-0_17. Acesso em: 24 de agosto de 2024.

HENRICO, Ivan. Método de interpolação ótimo para prever a batimetria da Baía do Saldanha. *Transactions in GIS*, v. 25, p. 1991-2009, 2021. Disponível em: https://doi.org/10.1111/tgis.12783. Acesso em: 09 de junho de 2024.

HODGE, Victoria J.; AUSTIN, Jim. A survey of outlier detection methodologies. *Artificial Intelligence Review*, v. 22, p. 85-126, 2004.

IGBINENIKARO, Osamuyimen; ADEKOYA, Olufemi; ETUKUDOH, Emmanuel. Revisão de técnicas modernas de levantamento batimétrico e seu impacto no desenvolvimento de energia offshore. *Revista de Ciência e Tecnologia de Engenharia*, v. 5, n. 4, 2024. Disponível em: https://doi.org/10.51594/estj.v5i4.1018. Acesso em: 18 de dezembro de 2024.

ILORI, Christopher; KNUDBY, Anders. Uma abordagem para minimizar o erro de correção atmosférica e melhorar a batimetria derivada de satélite baseada em física em um ambiente costeiro. *Remote Sensing*, v. 12, n. 17, p. 2752, 2020. Disponível em: https://doi.org/10.3390/rs12172752. Acesso em: 24 de janeiro de 2024.

IOFFE, Sergey; SZEGEDY, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv*, 2015. Disponível em: https://arxiv.org/abs/1502.03167. Acesso em: 24 de maio de 2024.

JANIESCH, Christian; ZSCHECH, Patrick; HEINRICH, Kai. Machine learning and deep learning. *Electronic Markets*, v. 31, p. 685-695, 2021.

JÖRGES, Christian; BERKENBRINK, Carsten; STUMPE, Björn. Previsão e reconstrução da altura das ondas oceânicas com base em dados batimétricos usando redes neurais LSTM. *Ocean Engineering*, 2021. Disponível em: https://doi.org/10.1016/J.OCEANENG.2021.109046. Acesso em: 25 de novembro de 2024.

KALOOP, Mosbeh; EL-DIASTY, Mohamed; HU, Jong; ZARZOURA, Farid. Redes Neurais Artificiais Híbridas para Modelagem de Batimetria em Águas Rasas via Imagens de Satélite. *IEEE Transactions on Geoscience and Remote Sensing*, v. 60, p. 1-11, 2022. Disponível em: https://doi.org/10.1109/TGRS.2021.3107839. Acesso em: 22 de setembro.

KUJAWA, Sebastian; NIEDBAŁA, Gniewko. Artificial Neural Networks in Agriculture. *Agriculture*, v. 11, n. 6, p. 497, 2021. Disponível em: https://doi.org/10.3390/AGRICULTURE11060497. Acesso em: 12 de agosto de 2024.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. *Nature*, v. 521, p. 436-444, 2015.

LIMA, João P.; SOUZA, Marcos A. Machine learning na detecção de anomalias em dados de levantamentos batimétricos. *Revista Brasileira de Geomática*, v. 12, n. 3, p. 45-58, 2020.

LI, Zheng; PENG, Zhong; ZHANG, Zhen; CHU, Ying; XU, Cheng; YAO, Shun; GARCÍA-FERNÁNDEZ, Álvaro; ZHU, Xiaoxiang; YUE, Yang; LEVERS, Andrew; ZHANG, Jian; JIANG, Jing. Explorando a batimetria moderna: uma revisão abrangente de dispositivos de aquisição de dados, precisão de modelos e técnicas de interpolação para mapeamento subaquático aprimorado. *Frontiers in Marine Science*, v. 10, 2023. Disponível em: https://doi.org/10.3389/fmars.2023.1178845. Acesso em: 09 de outubro de 2024..

MALHOTRA, Pankaj; VIG, Lovekesh; SHROFF, Gautam; AGARWAL, Puneet. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. 2016. Disponível em: arXiv:1607.00148. Acesso em: 12 de outubro de 2024.

MARINHA DO BRASIL. *Normas da Autoridade Marítima para Levantamentos Hidrográficos*. 2. ed. 2018.

MAYER, Larry; JAKOBSSON, Martin; ALLEN, G.; DORSCH, W.; FALCONER, R.; FERRINI, V.; GARDNER, J.; HALL, J.; KAMMERER, E.; MARKS, K.; MOHAMMADI, B.; MONAHAN, D.; MORISON, J.; MÜLLER, D.; OBENHOLZER, P.; PE'ERI, S.; POPE, A.; SMITH, W.; SNYDER, J.; STOCKWELL, R.; WEATHERALL, P. The Nippon Foundation—GEBCO seabed 2030 project: The quest to see the world's oceans completely mapped by 2030. *Geosciences*, v. 8, n. 2, p. 63, 2018.

MOHAMMADLOO, Tannaz; SNELLEN, Mirjam; SIMONS, Dick. Avaliação do desempenho do modelo batimétrico de predição de incerteza por ecobatímetro

multifeixe. *Applied Sciences*, v. 10, n. 13, p. 4671, 2020. Disponível em: https://doi.org/10.3390/app10134671. Acesso em: 14 de setembro de 2024.

MOLNAR, Christoph. *Interpretable Machine Learning*. Lulu.com, 2020.

OHI. *Norma da OHI – S-44: Especificações para Levantamentos Hidrográficos*. 6.1.0, 2020.

ORTEGA-FERNANDEZ, Iago; SESTELO, Marta; VILLANUEVA, Néstor. Redes neurais aditivas generalizadas explicáveis com treinamento de redes neurais independentes. *Statistics and Computing*, v. 34, 2023. Disponível em: https://doi.org/10.1007/s11222-023-10320-5. Acesso em: 30 de outubro de 2024.

PANG, Guansong; SHEN, Chao; CAO, Longbing; HENGEL, Anton van den. Deep Learning for Anomaly Detection: A Review. 2021. Disponível em: arXiv:2007.02500. Acesso em: 14 de agosto de 2024.

PIETROŁAJ, Michał; BLOK, Michał. Treinamento de rede neural com recursos limitados. *Scientific Reports*, v. 14, 2024. Disponível em: https://doi.org/10.1038/s41598-024-52356-1. Acesso em: 30 de junho de 2024.

QAMAR, Rizwan; ZARDARI, Babar. Artificial Neural Networks: An Overview. *Mesopotamian Journal of Computer Science*, 2023. Disponível em: https://doi.org/10.58496/mjcsc/2023/015. Acesso em: 21 de outubro de 2024.

QIAN, Y.; FORGHANI, M.; LEE, J.; FARTHING, M.; HESSER, T.; KITANIDIS, P.; DARVE, E. Aplicação de métodos de interpolação baseados em aprendizado profundo à batimetria nearshore. *arXiv*, 2020. Disponível em: https://arxiv.org/abs/2011.09707. Acesso em: 13 de janeiro de 2024.

RANGEL, Bruno R. *A aplicação de machine learning supervisionada para a predição de carbono orgânico no Atlântico Sul a partir de dados fisiográficos e taxas de sedimentação*. 2021. Dissertação (Mestrado) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2021.

SAUTER, Guido; FABBRI, Stefano; FRISCHKNECHT, Corinne; ANSELMETTI, Flavio; KREMER, Katrina. Uma abordagem sistêmica para o gerenciamento de incertezas em levantamentos batimétricos multifeixe repetitivos. *Environmental Modelling & Software*, v. 186, p. 106333, 2025. Disponível em: https://doi.org/10.1016/j.envsoft.2025.106333. Acesso em: 13 de janeiro de 2025.

SHAO, Yihao; DIETRICH, Felix; NETTELBLAD, Carl; ZHANG, Ce. Algoritmo de treinamento é importante para o desempenho do potencial de redes neurais. *The Journal of Chemical Physics*, v. 155, n. 20, p. 204108, 2021. Disponível em: https://doi.org/10.1063/5.0070931. Acesso em: 30 de janeiro de 2025.

SILVA, André; SIEBERT, Cristiano. DenseNet-DC: Optimizing DenseNet Parameters Through Feature Map Generation Control. *Revista de Informática Teórica e Aplicada*, v. 27, p. 25-39, 2020. Disponível em: https://doi.org/10.22456/2175-2745.98369. Acesso em: 14 de abril de 2024.

SILVA, Felipe R.; OLIVEIRA, Lucas A.; PEREIRA, Gustavo F. Redes neurais na otimização de processamento de dados batimétricos: um estudo comparativo. *Journal of Hydrographic Survey*, v. 8, n. 2, p. 112-130, 2021.

SRIVASTAVA, Nitish; HINTON, Geoffrey; KRIZHEVSKY, Alex; SUTSKEVER, Ilya; SALAKHUTDINOV, Ruslan. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, v. 15, n. 1, p. 1929-1958, 2014.

STEPHENS, David; SMITH, Andrew; REDFERN, Thomas; TALBOT, Andrew; LESSNOFF, Andrew; DEMPSEY, Kari. Using three dimensional convolutional neural networks for denoising echosounder point cloud data. *Applied Computing and Geosciences*, 2020. Disponível em: https://doi.org/10.1016/j.acags.2019.100016. Acesso em: 20 de abril de 2024.

SUN, Shuo; CHEN, Yifei; MU, Lin; LE, Yuan; ZHAO, Hong. Melhorando a Inversão Batimétrica em Águas Rasas por Meio de Transformação Não Linear e Redes Neurais Convolucionais Profundas. *Remote Sensing*, v. 15, n. 17, p. 4247, 2023. Disponível em: https://doi.org/10.3390/rs15174247. Acesso em: 25 de março de 2024.

TORROBA, Ignacio; BORE, Nils; FOLKESSON, John. Rumo a um Levantamento Batimétrico Autônomo em Escala Industrial. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), 2019. p. 6377-6382. Disponível em: https://doi.org/10.1109/IROS40897.2019.8968241. Acesso em: 14 de março de 2024.

WANG, Jian; TANG, Yang; JIN, Shaohua; BIAN, Guofeng; ZHAO, Xiaodong; PENG, Cong. Um método para levantamentos batimétricos multifeixe em águas desconhecidas com base no modo de profundidade constante do AUV. *Journal of Marine Science and Engineering*, v. 11, n. 7, p. 1466, 2023. Disponível em: https://doi.org/10.3390/jmse11071466. Acesso em: 13 de maio de 2024.

WŁODARCZYK-SIELICKA, Marta; BŁASZCZAK-BĄK, Wioleta. Processamento de Dados Batimétricos: A Fusão de Novos Métodos de Redução para Big Data Espacial. *Sensors*, v. 20, n. 21, 2020. Disponível em: https://doi.org/10.3390/s20216207. Acesso em: 14 de abril de 2024.

XIE, Yiping; BORE, Nils; FOLKESSON, John. Estimativa normal de rede neural e reconstrução batimétrica a partir de sonar de varredura lateral. *IEEE Journal of Oceanic Engineering*, v. 48, n. 1, p. 218-232, 2022. Disponível em: https://doi.org/10.1109/JOE.2022.3194899. Acesso em:20 de setembro de 2024.

XIE, Yiping; TRONI, Gabriele; BORE, Nils; FOLKESSON, John. Levantamento Batimétrico com Sonar de Imagem Utilizando Renderização de Volume Neural. *IEEE Robotics and Automation Letters*, v. 9, n. 9, p. 8146-8153, 2024. Disponível em: https://doi.org/10.1109/LRA.2024.3440843. Acesso em: 20 de dezembro de 2024.

YANG, Fan; XU, Feng; FAN, Meng; BU, Xiangwei; TU, Zhengkai; YAN, Xiaopeng. Um Método de Detecção Inteligente para Diferentes Tipos de Outliers em Nuvem de Pontos Batimétrica Multifeixe. *IEEE Transactions on Geoscience and Remote Sensing*, v. 60, p. 1-10, 2022. Disponível em: https://doi.org/10.1109/TGRS.2022.3209344. Acesso em: 20 de agosto de 2024.

ZHANG, Wei; LIU, Fang; NGUYEN, Cuong; YANG, Zhengyu; RAMASAMY, Suresh; FOO, Chuan-Sheng. Treinamento de redes neurais com regras de classificação para incorporar conhecimento de domínio. *Knowledge-Based*

Systems, v. 294, p. 111716, 2024. Disponível em: https://doi.org/10.1016/j.knosys.2024.111716. Acesso em: 14 de abril de 2024.

ZHONG, Jie; LIU, Xianwei; SHEN, Xin; JIANG, Lianhuan. Um algoritmo robusto para redução de ruído de fótons e estimativa batimétrica com base em dados do ICESat-2. *Remote Sensing*, v. 15, n. 8, p. 2051, 2023. Disponível em: https://doi.org/10.3390/rs15082051. Acesso em: 25 de março de 2024.

ZONG, Bo; SONG, Qi; MIN, Martin Renqiang; CHENG, Wei; LUMETTA, Christopher; CHEN, Wei; LIU, Yun. Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. 2018. Disponível em: arXiv:1806.02397. Acesso em: 13 de janeiro de 2025.

. .

APÊNDICES

APÊNDICE A - MODELAGEM E TREINAMENTO DE REDES NEURAIS COM KERAS

Neste apêndice, apresentamos uma visão detalhada sobre a modelagem de redes neurais utilizando a biblioteca *Keras*, abordando os seguintes tópicos:

- Arquitetura de redes neurais
- Funções de perda e métricas
- Otimização com o `Adam e outras estratégias
- Regularização com `Dropout` e `Batch Normalization`
- Código exemplo de implementação em Python

1. MODELAGEM E TREINAMENTO DE REDES NEURAIS COM KERAS

1.1 Introdução ao Keras

O Keras, desenvolvido por François Chollet, representa uma das bibliotecas mais intuitivas e poderosas para a construção e experimentação de modelos de aprendizado profundo (Chollet, 2017). Escrito em Python, o Keras fornece um método de alto nível para desenvolver e avaliar redes neurais, tornando o processo mais acessível a desenvolvedores e pesquisadores.

1.2 Fundação

O Keras foi criado com o objetivo de ser uma interface amigável e modular para redes neurais. Em contraste com outras bibliotecas de aprendizado profundo, que frequentemente se aprofundam em detalhes intrincados de otimização de algoritmos, o Keras busca simplificar a experiência do usuário (Chollet, 2017). Essa simplicidade, no entanto, não compromete sua versatilidade; ele é capaz de construir desde modelos simples de regressão até complexas redes generativas adversariais.

1.3 Características Principais

Modularidade: No Keras, quase tudo é tratado como um módulo. Isso inclui camadas neurais, funções de custo, otimizadores e até mesmo métricas de ativação. Esta modularidade permite aos usuários criarem modelos de aprendizado profundo combinando e personalizando módulos de forma flexível (Chollet, 2017).

Flexibilidade: O Keras suporta diferentes tipos de modelos, desde redes neurais totalmente conectadas, redes convolucionais (CNNs), até redes recorrentes (RNNs) (Goodfellow et al., 2016). Esta flexibilidade é essencial para atender às diversas necessidades dos problemas de aprendizado de máquina.

Integração com Backends: Uma das características mais distintas do Keras é sua capacidade de se integrar com várias bibliotecas de aprendizado profundo, como TensorFlow, Theano e Microsoft Cognitive Toolkit. Isso permite que os usuários escolham o backend que melhor atende às suas necessidades, seja em termos de performance ou funcionalidades específicas (Chollet, 2017).

2. CONSTRUÇÃO DE MODELOS NO KERAS

O desenvolvimento da aprendizagem profunda e, consequentemente, das redes neurais, é uma revolução na maneira como os computadores interpretam e trabalham com grandes volumes de dados. Dentro desse universo, a biblioteca Keras emergiu como uma das ferramentas mais populares, devido à sua simplicidade e flexibilidade. Com Keras, a construção e treinamento de modelos neurais complexos se tornaram mais acessíveis (Chollet, 2017).

2.1 A Natureza Modular do Keras

A característica mais distinta do Keras é sua modularidade. A ideia por trás dessa modularidade é que qualquer modelo de aprendizado profundo pode ser expresso como uma sequência ou um gráfico de camadas independentes e totalmente configuráveis (Chollet, 2017). Em termos matemáticos, podemos expressar um modelo M no Keras como:

$$M = f1(f2(f3(\dots fn(x) \dots))) \quad (1)$$

Onde:

M é a representação do modelo.

fn representa a n-ésima camada do modelo.

x é a entrada da camada inicial.

A.1.4. O Paradigma `Sequential`

Uma das abordagens mais comuns para a construção de modelos no Keras é usar o paradigma `Sequential`. Nesse paradigma, as camadas são empilhadas

uma após a outraem uma sequência linear (Chollet, 2017). Formalmente, podemos representar um modelo `Sequential` como:

$$M = f1 \circ f2 \circ f2 \circ \ldots \circ fn \qquad (2)$$

A natureza linear do modelo `Sequential` é benéfica para muitas tarefas, mas pode ser limitada quando se deseja criar arquiteturas mais complexas, como redes com múltiplas entradas ou saídas.

2.2 Camadas no Keras

No coração da construção de modelos no Keras estão as camadas. Cada camada é, em essência, um módulo de transformação que recebe uma entrada, a processa de alguma maneira e produz uma saída (Chollet, 2017). Algumas das camadas mais comuns no Keras incluem:

Dense (ou Fully Connected): Uma camada densa realiza a operação de ativação como mostrada na equação .

Convolutional: Essas camadas são essenciais para tarefas de processamento de imagem e operam através da aplicação de um filtro convolucional à entrada.

Recurrent: Camadas como LSTM e GRU, que são adequadas para sequências ou dados temporais.

Configuração e Compilação

Após definir a arquitetura do modelo, é vital configurar o processo de aprendizagem. Isso é feito através do método `compile`, que especifica o otimizador a ser usado, a função de perda e métricas adicionais para avaliação, como acurácia (Chollet, 2017).

A construção de modelos no Keras vai além da simples definição da arquitetura. É crucial entender os dados de entrada, pré-processá-los adequadamente e, eventualmente, fazer ajustes finos (tuning) nos hiperparâmetros do modelo para obter o melhor desempenho.

No núcleo da biblioteca Keras está a capacidade de construir modelos de aprendizado profundo. Um exemplo simplificado da construção de uma rede neural no Keras é:

Figura A.1 – Exemplo de modelo simplificado.

```
from keras.models import Sequential
from keras.layers import Dense

model = Sequential()
model.add(Dense(64, activation='relu', input_shape=(dimensão_de_entrada,)))
model.add(Dense(32, activation='relu'))
model.add(Dense(dimensão_de_saída, activation='softmax'))
```

Fonte: Elaborado pelo autor.

O modelo acima demonstra a natureza sequencial de construir redes neurais no Keras, adicionando camadas uma após a outra.

3. TREINAMENTO DE MODELOS COM KERAS

O treinamento de modelos de aprendizado de máquina, em particular redes neurais profundas, é uma tarefa computacionalmente intensiva. A biblioteca Keras, conforme descrito por Chollet (2017), tem sido amplamente reconhecida por sua simplicidade e flexibilidade, permitindo a rápida experimentação e eficaz treinamento de modelos neurais.

3.1 Compilação do Modelo

Antes de treinar um modelo, é essencial compilar o modelo escolhido, configurando o processo de aprendizado. A compilação é realizada usando o método `compile` que recebe três argumentos principais: um otimizador, uma função de perda e uma lista de métricas. A escolha adequada desses componentes é vital para o treinamento eficiente do modelo.

$$modelo.\,compile\,(optimizer, loss, metrics) \qquad (3)$$

Onde:

`**optimizer**` pode ser um dos otimizadores padrões do Keras, como 'adam' ou 'sgd'.

`**loss**` é a função objetivo que o modelo tentará minimizar. Exemplos comuns incluem 'mean_squared_error' para tarefas de regressão e 'categorical_crossentropy' para classificação.

`**metrics**` são usadas para monitorar o treinamento. Eles não afetam o treinamento em si (Chollet, 2017).

3.1 Alimentação de Dados

Para alimentar dados ao modelo, o método `fit` é usado. O Keras suporta a alimentação de dados como Numpy arrays ou como geradores de dados. O treinamento geralmente é realizado em lotes e por várias épocas. Uma época é uma única passagem por todo o conjunto de treinamento.

$$modelo.\ fit\ (x, y, epochs, batch\_size) \quad (4)$$

Aqui, `x` e `y` são os dados de entrada e saída, respectivamente. `epochs` define o número de vezes que o algoritmo de otimização passará pelo conjunto de dados e `batch_size` define o número de amostras após o qual o modelo é atualizado (Chollet, 2017).

3.2 Avaliação e Ajuste

A avaliação do modelo é tão crítica quanto seu treinamento. O Keras fornece o método `evaluate` que retorna à função de perda e as métricas para a avaliação.

$$loss, accuracy = modelo.\ evaluate\ (x\_test, y\_test) \quad (5)$$

Onde `x_test` e `y_test` são os dados de teste. O ajuste do modelo pode ser realizado ajustando-se os hiperparâmetros, escolhendo diferentes funções de perda ou otimizadores, ou mesmo alterando a arquitetura do modelo.

3.3 Regularização e Otimização

Prevenir o overfitting é um dos desafios mais significativos no treinamento de modelos neurais. Chollet (2017) discute técnicas como dropout, que é uma forma de regularização em que aleatoriamente certos neurônios são "desativados" durante o treinamento, ajudando a garantir que o modelo não se torne muito dependente de qualquer neurônio individual. Matematicamente, o dropout pode ser representado como:

$$h' = h \times Bernoulli(p) \quad (6)$$

Onde h' é a saída após o dropout, h é a saída original e p é a probabilidade de retenção.

Além disso, técnicas avançadas de otimização, como a utilização do otimizador Adam, que combina as vantagens dos algoritmos AdaGrad e RMSProp, podem acelerar o treinamento e alcançar melhores resultados (Chollet, 2017).

O treinamento eficaz de modelos neurais é uma combinação da escolha da arquitetura certa, dos hiperparâmetros apropriados, e da utilização de técnicas de regularização e otimização

3.4 Avaliação e Predição

Avaliar e prever são etapas essenciais no processo de desenvolvimento e aplicação ode modelos de aprendizado de máquina. Estas etapas permitem determinar o desempenho do modelo em dados não vistos anteriormente e sua capacidade de fazer previsões precisas em cenários do mundo real.

A avaliação de modelos envolve a utilização de métricas específicas para medir o desempenho do modelo em um conjunto de testes. Este conjunto de testes é tipicamente uma parcela dos dados que não foi usada durante o treinamento, garantindo uma avaliação imparcial do modelo (Hastie et al., 2009).

a) Métricas Comuns:

- **Erro Quadrático Médio (MSE)**: Utilizado principalmente para problemas de regressão. Mede a média dos quadrados dos erros entre previsões e valores reais.

$$MSE = \frac{1}{n} \sum_{i=1} (y_i - y_l)^2 \quad (7)$$

Onde yi é o valor real e y^7 é o valor previsto para a i-ésima observação, e n é o número total de observações.

- **Acurácia**: Amplamente utilizada em classificação, mede a proporção de previsões corretas em relação ao total.

$$Acurácia = \frac{Número\ de\ predições\ corretas}{Número\ total\ de\ predições} \quad (8)$$

- **ROC e AUC**: Para problemas de classificação binária, a Curva de Característica de Operação do Receptor (ROC) e a Área Sob a Curva (AUC) são métricas importantes. AUC é a área sob a curva ROC e indica o desempenho geral do modelo (Fawcett, 2006).

Após treinar e avaliar o modelo, a próxima etapa é usar esse modelo para fazer previsões. Isso pode ser feito para novos dados, ou seja, dados que o modelo nunca viu antes. O Keras, por exemplo, oferece uma função `predict` que torna essa tarefa simples(Chollet, 2017).

O objetivo é que estas previsões sejam o mais próximas possível da realidade, e a qualidade dessas previsões é determinada pelas métricas de avaliação mencionadas anteriormente.

É importante ressaltar que, em muitos cenários do mundo real, não basta apenas fazer uma previsão. A interpretação dessas previsões, o entendimento da incerteza associada e a tradução desses insights em ações concretas são tão cruciais quanto a própria previsão (Molnar, 2020).

Avaliar corretamente um modelo é crucial para entender sua eficácia e robustez. A predição, por outro lado, é o objetivo final do treinamento do modelo. Ambas as etapas são interdependentes e essenciais para garantir que o modelo não só funcione bem em um ambiente controlado, mas também tenha uma boa performance no mundo real.