

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA NAVAL

STEFANIE REZENDE CARVALHO

DESENVOLVIMENTO DE UMA REDE NEURAL MLP PARA ESTIMAR A
RESISTÊNCIA AO AVANÇO DE EMBARCAÇÕES PLANANTES UTILIZANDO O
MÉTODO SAVITSKY

Joinville
2022

STEFANIE REZENDE CARVALHO

DESENVOLVIMENTO DE UMA REDE NEURAL MLP PARA ESTIMAR A
RESISTÊNCIA AO AVANÇO DE EMBARCAÇÕES PLANANTES UTILIZANDO O
MÉTODO SAVITSKY

Trabalho apresentado como requisito para
obtenção do título de bacharel em Engenharia
Naval do Centro Tecnológico de Joinville da
Universidade Federal de Santa Catarina.

Orientador: Prof. Dr. Thiago Pontin Tancredi

Coorientador: M.Sc. Carsten Holst

Joinville
2022

STEFANIE REZENDE CARVALHO

DESENVOLVIMENTO DE UMA REDE NEURAL MLP PARA ESTIMAR A
RESISTÊNCIA AO AVANÇO DE EMBARCAÇÕES PLANANTES UTILIZANDO O
MÉTODO SAVITSKY

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de bacharel em Engenharia Naval, na Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Joinville (SC), 13 de Dezembro de 2022.

Banca Examinadora:

Prof. Dr. Thiago Pontin Tancredi
Orientador e Presidente da Banca

Prof. Dr. Lucas Weihmann
Membro da Banca
Universidade Federal de Santa Catarina

Prof. Dr. Pablo Andretta Jaskowiak
Membro da Banca
Universidade Federal de Santa Catarina

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me dado a oportunidade de viver, ser saudável e por me conceder a honra de ter pais extraordinários, a quem sou eternamente grata. Toda a minha trajetória jamais aconteceria sem a coragem, força e empenho de vocês.

À minha mãe, Klebiana, eu agradeço o apoio incondicional, responsável por me manter firme na busca pelos meus objetivos pessoais e profissionais.

Ao meu pai, Argeu, agradeço o esforço para me proporcionar a melhor educação possível e por sempre ter sido o meu exemplo moral e ético que levarei para o resto da vida.

Ao meu marido, Eduardo, agradeço por aceitar trilhar esta nova jornada comigo, longe de casa e das pessoas que mais amamos. Também sou grata por seu apoio e paciência neste que foi sem dúvida o trabalho mais árduo e desafiador de toda a minha graduação.

À minha avó, Tamara, por ter sido a minha segunda mãe e ao tio Zuga, cujos conselhos e dicas para que eu aprendesse inglês foram muito valiosos. Eu os apliquei ao estudar alemão.

Ao meu tio Márcio (in memoriam), cujo carinho e afeto moldam as memórias da minha infância, fantástica e inesquecível.

Ao professor Maurício de Campos Porath por acreditar em mim e me selecionar para participar do estágio que, definitivamente, mudaria o rumo da minha vida.

Ao meu orientador, Prof. Thiago Pontin Tancredi, por me sugerir trabalhar com redes neurais artificiais, tema que junto à programação constituem um novo aspecto, disruptivo, na minha formação acadêmica.

E, por fim, à Universidade Federal de Santa Catarina pelo ensino gratuito e a todos os seus professores que de alguma forma contribuíram para a minha formação. Meu eterno agradecimento e respeito a vocês.

"Neurons that fire together wire together."

Donald O. Hebb

RESUMO

Algoritmos de *Machine Learning* e redes neurais artificiais representam um marco para as tecnologias digitais atuais. Entretanto, a aplicabilidade desses modelos não se restringe unicamente a problemas de reconhecimento de imagens ou síntese de fala, sendo fortemente aplicados na geração de modelos preditivos de difícil formulação, mas com histórico conhecido. Em se tratando da Engenharia Naval, um dos problemas que se encaixa nesse contexto é justamente a estimativa da resistência ao avanço de uma embarcação. Embora diversos trabalhos proponham o uso de algoritmos de *Machine Learning* para o cálculo da resistência ao avanço, não foram encontradas iniciativas que apliquem tais técnicas à estimativa da resistência ao avanço de embarcações planantes. Nesse sentido, este trabalho apresenta o desenvolvimento de redes neurais artificiais destinadas à estimativa da resistência ao avanço de embarcações planantes cujo conjunto de treinamento é gerado aplicando-se o clássico método Savitsky. Para tanto, levantou-se os dados referentes a 54 cascos planantes e utilizou-se uma implementação conhecida do método de Savitsky para estimar a resistência ao avanço considerando sete valores de velocidade. O algoritmo proposto foi desenvolvido em Python e utiliza bibliotecas sabidamente robustas e eficientes para o tratamento dos dados, definição da topologia das redes, treinamento, validação e plotagem de resultados. Visando determinar os parâmetros topológicos da rede que minimizam o erro quadrático médio, foram explorados 12 modelos neurais, os quais foram treinados por 100 ciclos em lotes de 15 pontos, sempre utilizando um conjunto de treinamento que corresponde a 70% dos dados disponíveis. Os resultados mostraram que todas as topologias foram capazes de identificar padrões, permitindo estimar a resistência ao avanço de um casco planante com diferentes graus de precisão, com a inexplicável exceção da rede M8, cujo comportamento não pôde ser compreendido. Embora o modelo M12 tenha apresentado o menor erro quadrático médio (4,30) em relação ao conjunto de teste, considera-se que o modelo M4, obteve um desempenho melhor, pois, além de apresentar o segundo menor erro quadrático médio (6,37), conseguiu estimar a resistência ao avanço utilizando uma quantidade menor de parâmetros de entrada do que o modelo M12. Por fim, os resultados mostraram que o parâmetro que mais influenciou no desempenho das redes estudadas foi o número de camadas internas do modelo, sendo que as redes com duas camadas internas apresentaram sempre um erro quadrático médio inferior às redes com apenas uma camada interna.

Palavras-chave: Redes Neurais. *Machine Learning*. Embarcações Planantes. Resistência ao avanço. Método Savitsky.

ABSTRACT

Machine Learning algorithms and artificial neural networks represent a milestone for current digital technologies. However, the applicability of these models is not restricted solely to problems of image recognition or speech synthesis, being strongly applied in the generation of predictive models of difficult formulation, but with known history. When it comes to Naval Engineering, one of the problems that fits in this context is precisely the estimation of the resistance. Although several works propose the use of Machine Learning algorithms for the calculation of resistance, no initiatives have been found that apply such techniques to the estimation of the resistance of planing ships. In this sense, this work presents the development of artificial neural networks to estimate the resistance of planing ships whose training set is generated by applying the classic Savitsky method. For this purpose, data from 54 planing hulls was collected and a known implementation of Savitsky's method was used to estimate the resistance considering seven speed values. The proposed algorithm was developed in Python and uses libraries known to be robust and efficient for data processing, network topology definition, training, validation, and result plotting. In order to determine the topological parameters of the network that minimize the mean square error, 12 neural models were explored, which were trained for 100 epochs in batches of 15 points, always using a training set that corresponds to 70% of the available data. The results showed that all topologies were able to identify patterns, allowing estimation of the resistance of a planing hull with varying degrees of accuracy, with the unexplained exception of the M8 network, whose behavior could not be understood. Although the M12 model presented the lowest mean square error (4.30) with respect to the test set, the M4 model, is considered to have performed better because, in addition to presenting the second lowest mean square error (6.37), it was able to estimate the resistance using a smaller amount of input parameters than the M12 model. Finally, the results showed that the parameter that most influenced the performance of the networks studied was the models' number of hidden layers, and the networks with two hidden layers always presented a lower mean square error than networks with only one.

Keywords: Neural Networks. Machine Learning. Planing Vessels. Drag. Savitsky method.

LISTA DE FIGURAS

Figura 1 – Programação tradicional versus Machine Learning	17
Figura 2 – Fases de um projeto naval	22
Figura 3 – Matriz de influência	23
Figura 4 – Exemplo de espiral de projeto	24
Figura 5 – Diagrama de corpo livre para uma embarcação em planeio	25
Figura 6 – Principais dimensões de uma embarcação planante (vista lateral à esquerda) e ângulo de deadrise (vista frontal à direita)	26
Figura 7 – Componentes da resistência ao avanço	27
Figura 8 – Sistema de onda primário (a) e sistema de onda secundário (b)	29
Figura 9 – Métodos e exemplos para estimativa da resistência ao avanço	30
Figura 10 – Distribuição de pressão em uma placa plana	31
Figura 11 – Modelo não linear de um neurônio	34
Figura 12 – Comportamento da função sigmoide (a) e função tangente hiperbólica (b)	36
Figura 13 – Rede neural <i>Feedforward</i> com múltiplas camadas	37
Figura 14 – Abordagem para a classificação de padrões	38
Figura 15 – Metodologia empregada	40
Figura 16 – Função de ativação ReLU	42
Figura 17 – Fluxograma de treinamento e avaliação das redes neurais	43
Figura 18 – Bibliotecas importadas	46
Figura 19 – Função <i>readInputParams()</i>	47
Figura 20 – Arquivo <i>params.csv</i>	47
Figura 21 – Função <i>preparaDataset()</i>	48
Figura 22 – Função <i>splitDataset()</i>	48
Figura 23 – Função <i>buildTopology()</i>	49
Figura 24 – Função <i>trainModel()</i>	50
Figura 25 – Função <i>perGenerationLoss()</i>	51
Figura 26 – Execução do código	52
Figura 27 – Repositório <i>Resistance Prediction</i>	53
Figura 28 – Curvas de aprendizado - M1 a M6	55
Figura 29 – Curvas de aprendizado - M7 a M12	56
Figura 30 – Erro quadrático médio do conjunto de teste por modelo	59
Figura 31 – Distribuição espacial dos EQM dos modelos neurais estudados	61
Figura 32 – Posição espacial: modelo M12 (a) e modelo M1 (b)	61
Figura 33 – Curva de resistência por velocidade para Schaefer V33	62

Figura 34 – Curva de resistência por velocidade para BG54	63
Figura 35 – Curva de resistência por velocidade para Azimut 75	63

LISTA DE QUADROS

Quadro 1 – Coeficientes e parâmetros presentes no modelo de Savitsky	32
Quadro 2 – Funções de ativação mais comuns utilizadas em problemas de engenharia naval e oceânica	35

LISTA DE TABELAS

Tabela 1 – Modelos de redes neurais propostos	41
Tabela 2 – Grupos de parâmetros navais de entrada	41
Tabela 3 – Análise estatística do conjunto de dados utilizado no desenvolvimento	45
Tabela 4 – Principais funcionalidades das bibliotecas utilizadas	46
Tabela 5 – Número de conexões e de ciclos de treinamento até à estabilização do treinamento de cada modelo	57
Tabela 6 – EQM para o último ponto das curvas de treinamento e validação e valor referente ao espaço de generalização (EG)	58
Tabela 7 – Análise paramétrica dos modelos	60
Tabela 8 – Identificação das embarcações do conjunto de dados	70
Tabela 9 – Conjunto de dados (Parte I)	71
Tabela 10 – Conjunto de dados (Parte II)	72
Tabela 11 – Conjunto de dados (Parte III)	73
Tabela 12 – Conjunto de dados (Parte IV)	74
Tabela 13 – Conjunto de dados (Parte V)	75
Tabela 14 – Conjunto de dados (Parte VI)	76
Tabela 15 – Resistência calculada por M4 (Parte I)	77
Tabela 16 – Resistência calculada por M4 (Parte II)	78
Tabela 17 – Resistência calculada por M12 (Parte I)	79
Tabela 18 – Resistência calculada por M12 (Parte II)	80
Tabela 19 – Resistência calculada por M1 (Parte I)	81
Tabela 20 – Resistência calculada por M1 (Parte II)	82

LISTA DE SÍMBOLOS

B	Boca da embarcação medida entre as quinas principais no espelho de popa
B_{\max}	Boca moldada máxima da embarcação
C_F	Coefficiente de atrito turbulento de Schoenherr
C_{L0}	Coefficiente de sustentação para $\beta=0$
$C_{L\beta}$	Coefficiente de sustentação para $\beta \neq 0$
C_p	Distância do centro de pressão da popa transom
C_v	Coefficiente de velocidade
D	Arrasto total
D_f	Arrasto de atrito
f	Distância entre a linha de operação do propulsor até o VCG
F_r	Número de Froude
g	Aceleração da gravidade
h	Imersão da popa transom
L	Comprimento da embarcação
L_{CG}	Centro longitudinal de gravidade em relação à popa
L_c	Comprimento molhado até a quina principal
L_k	Comprimento molhado até a quilha
L_p	Distância da popa transom até o ponto de interseção do vetor de força hidrodinâmica com a quilha
LOA	Comprimento total da embarcação
LWL	Comprimento da linha d'água
L_x	Distância até a meia nau
R_A	Coefficiente de correlação modelo-navio

R_{APP}	Resistência dos apêndices
R_B	Resistência adicional do bulbo
R_e	Número de Reynolds para superfície planante
R_F	Resistência de atrito
R_n	Número de Reynolds
R_T	Resistência total
R_{TR}	Resistência adicional do espelho de popa submerso
R_V	Resistência de pressão viscosa
R_W	Resistência de onda
S	Área molhada do casco
T	Empuxo
T_{max}	Calado máximo
V	Velocidade
$V_{cruzeiro}$	Velocidade de cruzeiro da embarcação
V_{max}	Velocidade máxima da embarcação
VCG	Centro vertical de gravidade em relação à quilha
V_1	Velocidade média no fundo da superfície
Z	Pontal moldado
$(1 + k)$	Fator de forma
β	Ângulo de <i>deadrise</i>
β_T	Ângulo de <i>deadrise</i> na popa transom
β_x	Ângulo de <i>deadrise</i> à meia nau
Δ	Deslocamento
ε	Ângulo entre a quilha e a linha do propulsor
θ	Ângulo entre a quilha e a quina principal
λ	Razão comprimento-boca molhada

ν	Viscosidade cinemática do fluido
ρ	Densidade do fluido
τ	Ângulo de trim: ângulo entre a quilha e a superfície da água

SUMÁRIO

1	INTRODUÇÃO	17
1.1	OBJETIVOS	19
1.1.1	Objetivo Geral	19
1.1.2	Objetivos Específicos	19
1.2	ESTRUTURA DO TRABALHO	19
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	PROJETO NAVAL	21
2.1.1	Matriz de influência	22
2.1.2	Espiral de Evans	23
2.2	EMBARCAÇÕES PLANANTES	24
2.3	RESISTÊNCIA AO AVANÇO	27
2.3.1	Conceito geral	27
2.3.2	Métodos para estimar a resistência ao avanço	30
2.3.2.1	Método Savitsky	31
2.4	REDES NEURAIS	33
2.4.1	Unidade neural	33
2.4.2	Funções de ativação	34
2.4.3	Topologia da rede neural	36
2.4.4	Aprendizado e treinamento da rede neural	37
3	METODOLOGIA	40
4	DESENVOLVIMENTO	44
4.1	CONJUNTO DE DADOS	44
4.2	REDES NEURAIS	46
4.2.1	Importação de bibliotecas	46
4.2.2	Definição de funções	47
4.2.3	Execução do código	51
4.2.4	Repositório <i>Resistance Prediction</i>	52
5	RESULTADOS E DISCUSSÕES	54
5.1	CURVAS DE APRENDIZADO	54
5.2	ANÁLISE GERAL DOS MODELOS	58
5.3	ANÁLISE PARAMÉTRICA	59
5.4	CURVAS DE RESISTÊNCIA POR VELOCIDADE	62

6	CONCLUSÕES	65
	REFERÊNCIAS	67
	APÊNDICE A	70
	APÊNDICE B	71
	APÊNDICE C	72
	APÊNDICE D	73
	APÊNDICE E	74
	APÊNDICE F	75
	APÊNDICE G	76
	APÊNDICE H	77
	APÊNDICE I	78
	APÊNDICE J	79
	APÊNDICE K	80
	APÊNDICE L	81
	APÊNDICE M	82

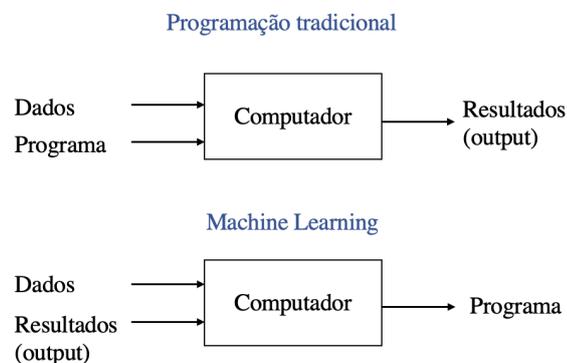
1 INTRODUÇÃO

O termo Inteligência Artificial (IA) é empregado genericamente para classificar quaisquer programas e/ou equipamentos que reproduzem aspectos da inteligência humana como, por exemplo, a tomada de decisões. Dentro do termo IA estão inseridos os conceitos de *Machine Learning*, *Neural Networks* e *Deep Learning*, sendo que esses dois últimos se diferenciam pela quantidade ou profundidade das camadas internas existentes no modelo neural (KAVLAKOGLU, 2020).

O conceito de *Machine Learning* (ML) pode ser entendido como o uso de algoritmos para analisar dados, identificar padrões e, finalmente, realizar estimativas do comportamento do sistema diante de novos parâmetros de entrada, ou ainda, ajustar variáveis de um processo buscando aperfeiçoar o resultado obtido (BROWNLEE, 2020).

Ainda segundo Brownlee (2020), diferentemente da programação tradicional que tem como saída (output) os resultados de uma análise, em ML a saída é o programa em si, haja vista que os resultados conhecidos fazem parte dos parâmetros de entrada, sendo inseridos juntamente com os demais dados referentes ao processo (Figura 1).

Figura 1 – Programação tradicional versus Machine Learning



Fonte: Adaptado de Brownlee (2020).

Dados de 2021 mostram que 56 % das empresas atuais utilizam ML em pelo menos uma de suas funções de negócio. Os resultados indicam ainda que o uso de IA cresceu mais nas empresas sediadas em economias emergentes, onde, em virtude da pandemia de COVID-19, a aplicação de tecnologias digitais capazes de automatizar processos de tomada de decisão, passou a ser entendida como uma nova necessidade para os atuais modelos de negócio (MCKINSEY, 2021).

De acordo com Yang et al. (2021), algoritmos de IA são amplamente empregados em problemas de reconhecimento de imagens e para a síntese da fala. Porém esses algoritmos vêm sendo cada vez mais aplicados em problemas

de hidrodinâmica e mecânica dos fluidos, para os quais a análise tem elevado custo computacional.

No âmbito da engenharia naval, dentre os problemas de hidrodinâmica mais usuais, a estimativa da resistência ao avanço de um casco merece destaque, pois é um importante indicador de desempenho de uma embarcação, além de ser um parâmetro fundamental para a definição do sistema propulsivo a bordo (MOLLAND, 2008).

Segundo Molland (2008), os métodos estabelecidos na literatura para a estimativa deste parâmetro variam desde fórmulas empíricas e séries sistemáticas até o desenvolvimento de modelos para ensaios em tanque de provas e simulações numéricas (CFD).

No entanto, no caso de projetos não convencionais, os diferentes modelos empíricos podem ser pouco eficazes para a estimativa adequada da resistência ao avanço, ou ainda pouco eficientes para avaliar modificações locais na geometria do casco, fazendo com que, nesses casos, apenas as técnicas numéricas (CFD) sejam capazes de realizar estimativas confiáveis da resistência ao avanço.

Paradoxalmente, assim como descrito por Leme (2021) as poucas ferramentas que implementam técnicas numéricas (CFD) para a estimativa da resistência ao avanço de uma embarcação apresentam um elevado custo computacional, interface pouco amigáveis e/ou elevado custo financeiro.

Visando mitigar esses problemas, autores como Yang et al. (2021), Cepowski (2020), Néia (2016) e Couser et al. (2004) utilizaram algoritmos de ML para estimar a resistência ao avanço total ou parcial de embarcações deslocantes. Os resultados foram promissores e sugerem a possibilidade do treinamento de redes neurais para o cálculo da resistência ao avanço de um casco, minimizando o número de análises numéricas realizadas durante o projeto de uma embarcação.

Embora diversos trabalhos encontrados na literatura abordem o uso de algoritmos de ML para estimar a resistência ao avanço, em geral, o objeto de estudo refere-se a embarcações deslocantes, gerando uma lacuna de conhecimento no que se refere à aplicação de técnicas de IA na estimativa da resistência ao avanço de embarcações planantes.

Diante desse cenário, este trabalho apresenta o desenvolvimento de um modelo de IA para estimar a resistência ao avanço de embarcações planantes por meio do uso de redes neurais implementadas em Python. Os dados de resistência utilizados no treinamento do modelo foram obtidos com auxílio da planilha de Tweedie (2011) que, por sua vez, utiliza o método empírico de Savitsky (1964).

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver uma rede neural em Python capaz de estimar a resistência ao avanço de embarcações planantes cujos dados de referência foram calculados utilizando-se o método de Savitsky (1964).

1.1.2 Objetivos Específicos

A fim de atingir o objetivo geral, propõem-se os seguintes objetivos específicos:

- a. Levantar e agrupar dados de no mínimo 50 cascos planantes, disponíveis na literatura;
- b. Calcular as resistências ao avanço das embarcações estudadas utilizando o método de Savitsky (1964) por meio da planilha desenvolvida por Tweedie (2011);
- c. Gerar o conjunto de dados completo reunindo as informações obtidas nos itens a e b;
- d. Planejar diferentes modelos de redes neurais (topologias) com base na literatura, variando tanto os parâmetros da rede quanto os parâmetros navais;
- e. Desenvolver e treinar os modelos de redes neurais estabelecidos no item d, utilizando o conjunto de dados elaborado no item c;
- f. Identificar dentre os modelos gerados aquele(s) que apresentam a melhor capacidade de generalização e o menor erro quadrático médio em relação ao conjunto de teste;
- g. Avaliar a influência dos parâmetros de rede utilizados no desempenho do modelo para a previsão de novos resultados;
- h. Realizar o estudo de caso considerando três embarcações do conjunto de dados utilizando os modelos de redes neurais com melhor e pior desempenho;
- i. Disponibilizar no *GitHub* o código de rede neural desenvolvido, visando facilitar a elaboração de trabalhos futuros.

1.2 ESTRUTURA DO TRABALHO

O desenvolvimento realizado é descrito em cinco capítulos. Definido o contexto do trabalho, no Capítulo 2 é apresentada a fundamentação teórica que embasa o desenvolvimento realizado. Dentre os assuntos abordados, destacam-se o projeto naval, as embarcações planantes, a resistência ao avanço, o método de Savitsky e, por último, as redes neurais artificiais.

Já o Capítulo 3 destina-se a descrever a metodologia utilizada, estabelecendo o processo de trabalho, além das técnicas e ferramentas utilizadas no desenvolvimento

proposto. Descreve-se também o planejamento dos modelos de redes neurais (topologias), bem como a divisão do conjunto de dados e o treinamento das redes.

A seguir, o desenvolvimento do código, desde a importação das bibliotecas, manipulação dos dados, definição de funções até a execução do algoritmo completo é descrito no Capítulo 4.

No Capítulo 5 são apresentados e discutidos os resultados obtidos, destacando-se as curvas de treinamento e avaliação geral, as quais são analisadas em função do erro quadrático médio. Por fim são analisados os 12 modelos estudados, buscando estabelecer a influência da topologia da rede na capacidade de predição do modelo.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos fundamentais referentes à execução de projetos navais, à distinção entre embarcações deslocantes e planantes, à estimativa da resistência ao avanço, ao entendimento do método de Savitsky e à modelagem de redes neurais. Tais conceitos embasam o desenvolvimento realizado neste trabalho e que será descrito no Capítulo 4.

2.1 PROJETO NAVAL

Usualmente um projeto naval é dividido em fases, pois a natureza do trabalho feito, as habilidades requeridas, o número de participantes envolvidos, as ferramentas utilizadas, o nível de detalhamento e outros atributos do processo, mudam ao longo do desenvolvimento do projeto (LAMB, 2003).

Nesse sentido, os esforços envolvidos durante o processo podem ser facilitados na medida em que novas possibilidades de revisão, preparo e planejamento de etapas surgem como resultado direto da técnica de gerenciamento de projetos. De acordo com Lamb (2003), as etapas de projeto podem ser diferenciadas de acordo com o nível de detalhamento envolvido, sendo classificadas em:

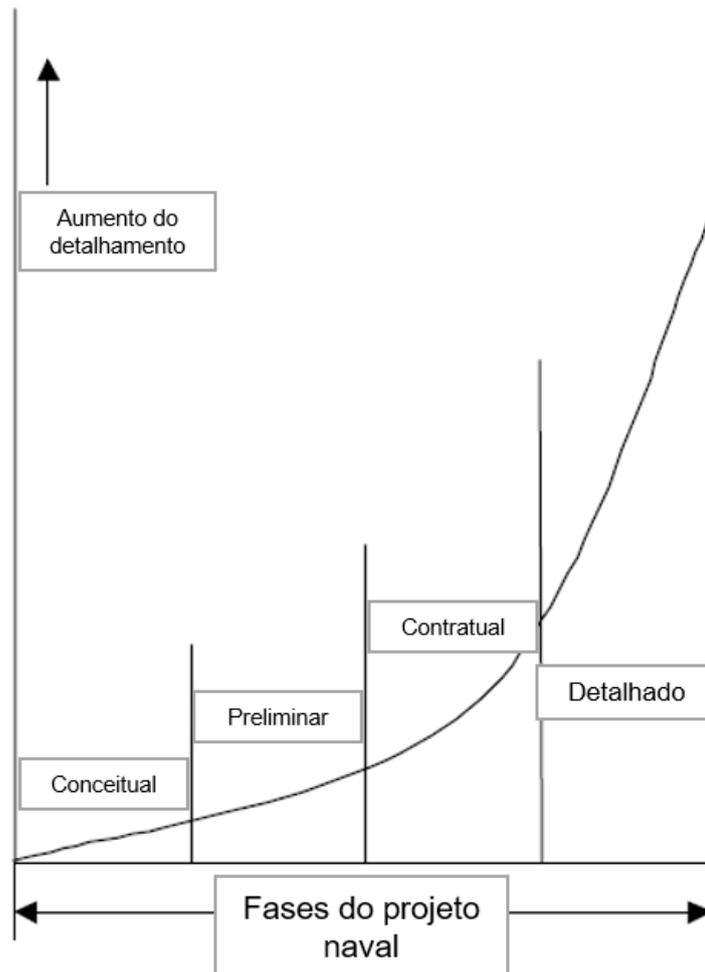
- Projeto Conceitual;
- Projeto Preliminar;
- Projeto Contratual;
- Projeto Detalhado.

Enquanto a etapa conceitual lida com o estudo da missão, levantamento de requisitos do armador, estimativa inicial das dimensões principais e busca de soluções, a etapa preliminar prevê uma estimativa elaborada desses atributos, ao mesmo tempo em que apresenta os cálculos iniciais necessários para o sistema propulsivo. Tais etapas representam o estágio inicial do projeto naval, no qual as principais características técnicas e econômicas do navio são determinadas (PAPANIKOLAOU, 2014).

Quanto ao projeto contratual e ao detalhado, o primeiro objetiva propiciar as informações necessárias para o estudo referente à viabilidade econômica do projeto, enquanto o segundo visa reunir as informações cruciais para a construção da embarcação (LAMB, 2003).

Na Figura 2 as etapas de projeto são dispostas em um gráfico bidimensional, o qual ilustra a evolução do grau de detalhamento de solução em função das fases do projeto.

Figura 2 – Fases de um projeto naval



Fonte: Adaptado de Lamb (2003, p. 5–7).

É possível notar que à medida que o projeto naval é desenvolvido, o nível de detalhamento presente em cada etapa aumenta significativamente, indicando uma definição mais precisa dos atributos funcionais da embarcação, o que, geralmente, implica em um processo de sequenciamento e constante revisão das atividades de projeto.

2.1.1 Matriz de influência

Entre as diferentes técnicas utilizadas no sequenciamento das atividades de um projeto, a matriz de influência costuma, apesar da simplicidade, fornecer bons resultados práticos (BRUNER, 2019).

Ainda segundo Bruner (2019), uma matriz de influência é elaborada visando entender os principais atributos de uma embarcação e seu grau de prioridade

específico associado ao projeto em desenvolvimento. Nessa técnica, os atributos funcionais são enumerados tanto nas linhas quanto nas colunas da matriz, sendo que a cada interseção, uma nota é atribuída indicando o grau de influência que um atributo exerce sobre o outro. A partir do somatório das notas atribuídas, é possível definir o grau de prioridade de cada atributo no projeto de uma embarcação (Figura 3).

Figura 3 – Matriz de influência

MATRIZ DE INFLUÊNCIA	Dimensões e Coef.	Resistência ao Avanço	Instalação propulsora	Velocidade Econômica	Arranjo 2D	Desenho do Casco	Pesos e Centros	Estabilidade	Projeto Estrutural	Manobrabilidade	Arranjo 3D	TOTAL	Ordem de Prioridade		
Dimensões e Coef.												0	2	Dimensões e Coef.	0
Resistência ao Avanço	1											1	3	Resistência ao Avanço	1
Instalação propulsora	1	1		1		1						4	4	Instalação propulsora	4
Velocidade Econômica	1	1										2	5	Velocidade Econômica	2
Arranjo 2D	1		1									2	6	Arranjo 2D	2
Desenho do Casco	1											1	7	Desenho do Casco	1
Pesos e Centros	1		1		1							3	8	Pesos e Centros	3
Estabilidade	1					1	1					3	9	Estabilidade	3
Projeto Estrutural	1		1		1	1	1					5	10	Projeto Estrutural	5
Manobrabilidade	1	1	1	1	1	1	1				1	7	11	Manobrabilidade	7
Arranjo 3D	1	1	1		1	1	1					6	12	Arranjo 3D	6
	1	1	1		1	1	1					6	13	Manobrabilidade	7

Fonte: Bruner (2019, p. 18).

Além de indicar a ordem de atributos a ser calculada, a matriz de influência ajuda na construção da Espiral de Evans, elemento fundamental para o desenvolvimento de projetos navais (WATSON, 1998).

2.1.2 Espiral de Evans

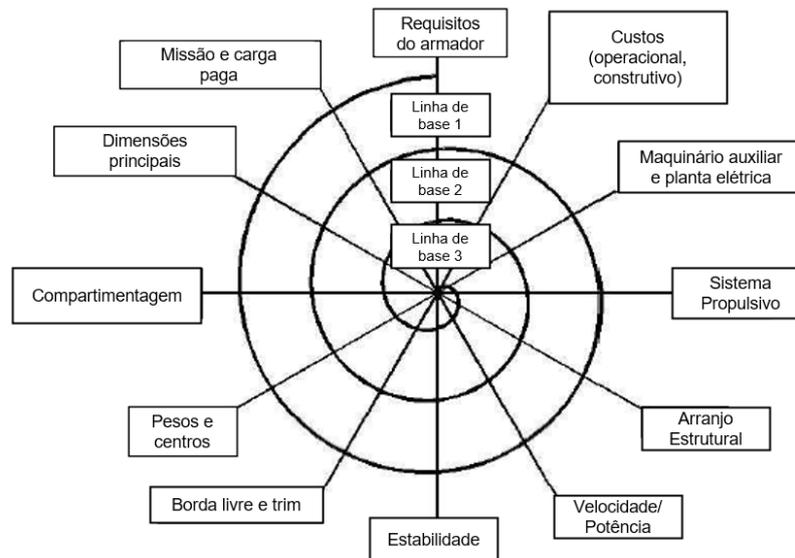
Visando sistematizar o projeto naval, iterativo em sua essência, a abordagem proposta por Evans (1959) representa o processo em uma única figura (Figura 4), a qual dispõe os principais atributos funcionais de uma embarcação na ordem de execução a ser seguida durante o projeto. Nesse sentido, a matriz de influência é uma importante técnica capaz de auxiliar na construção da espiral de projeto.

Essa abordagem permite ao projetista embasar o desenvolvimento de um projeto naval, podendo priorizar os atributos fundamentais à missão da embarcação, ou ainda, identificar os elementos que exigem um maior grau de detalhamento do projeto para serem calculados.

No entanto, nenhum projeto está imune a problemas e dificuldades, que, em muitos casos, decorrem da falta de informação disponível naquela etapa do projeto. Nesses casos, podem ocorrer mudanças na ordem prevista pela espiral de projeto, o que significa dizer que, eventualmente, alguns atributos podem ser calculados antes de outros, ou ainda em paralelo (LAMB, 2003).

Além disso, é importante salientar que, dado o caráter interdisciplinar e interdependente de um projeto naval, é natural que cada etapa seja revisitada inúmeras vezes, exigindo processos eficientes e com baixo custo computacional, especialmente nas fases iniciais do projeto.

Figura 4 – Exemplo de espiral de projeto



Fonte: Adaptado de Lamb (2003, p. 5-2).

Observa-se no exemplo de espiral de projeto mostrado na Figura 4 que o atributo de entrada aplicado (requisitos do armador) costuma ser fixo, uma vez que os projetos navais são construídos em cima de premissas, limitações e desejos estabelecidos pelo armador, além das restrições impostas pelas regras e normas vigentes (PAPANIKOLAOU, 2014).

2.2 EMBARCAÇÕES PLANANTES

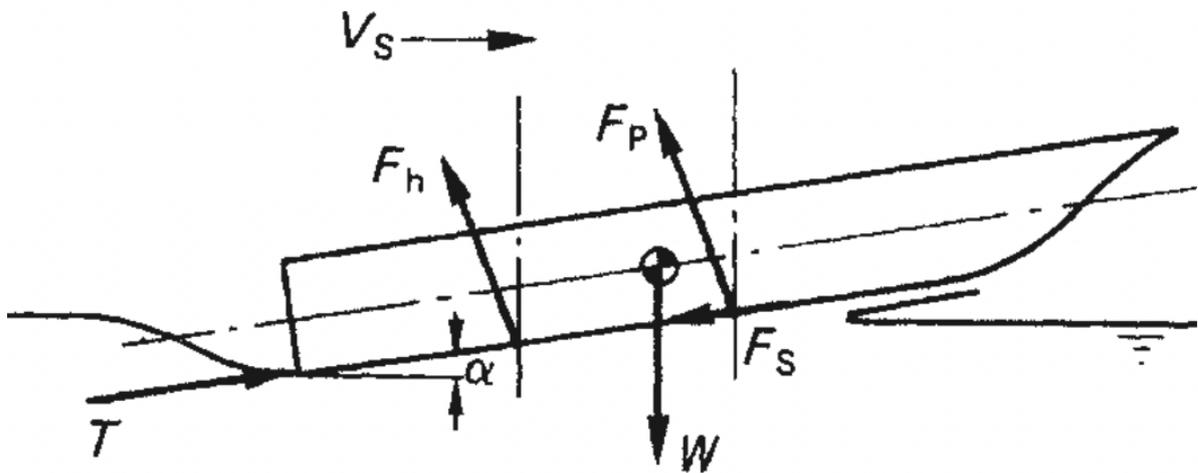
Conforme já mencionado, um projeto naval é realizado para atender aos requisitos específicos definidos pelo armador, além de obedecer às normas estabelecidas pela sociedade classificadora e pelas autoridades marítimas; apresentando características funcionais, forma do casco, sistema propulsivo e arranjo estrutural adequados à missão da embarcação.

É justamente a missão da embarcação que define as características mais fundamentais do projeto. Ou seja, um navio que tem como objetivo transportar carga, deve, necessariamente, apresentar capacidade interna suficiente para justificar o custo de operação da embarcação (MOLLAND, 2008).

Nesse sentido, diz-se que a missão de uma embarcação de recreio é dada pela sua capacidade de entreter a tripulação enquanto apresenta atributos de conforto e velocidade satisfatórios. No entanto, para Dias Neto, Onety e Tancredi (2021), conforto e velocidade são atributos conflitantes, os quais devem ser otimizados com base nos desejos do armador.

Segundo Papanikolaou (2014), as embarcações planantes se diferem das deslocantes por apresentarem apenas uma parte do peso suportado pela força hidrostática. Nesse caso, o restante do peso será suportado por uma segunda força, chamada de sustentação, a qual tem natureza hidrodinâmica (Figura 5).

Figura 5 – Diagrama de corpo livre para uma embarcação em planeio



Fonte: Molland (2008, p. 216).

Onde:

- V_S - Velocidade;
- T - Empuxo do propulsor;
- W - Peso da embarcação;
- F_S - Resistência ao avanço;
- F_h - Força hidrostática;
- F_p - Força hidrodinâmica.

Por esse motivo, o projeto de embarcações planantes apresenta, desde o início, princípios físicos distintos daqueles vistos em embarcações deslocantes, para as quais, o Princípio de Arquimedes é válido (FALTINSEN, 2005).

De acordo com Kukner e Yasa (2011), existem dois campos de pressão atuando no casco de embarcações planantes, o hidrostática e o hidrodinâmica. O primeiro refere-se à flutuação e é proporcional ao volume de água deslocado pela parte submersa do

casco. Já o segundo, depende do fluxo ao redor do casco, sendo proporcional à raiz da velocidade relativa entre o casco e o escoamento.

Considera-se ainda que uma embarcação está em regime de planeio quando o seu número de Froude (Equação 1) é maior que 1,2. Porém, a delimitação entre os regimes de pré-planeio e planeio não é exata e o número de Froude igual a 1 também costuma ser utilizado para definir o início do planeio (FALTINSEN, 2005).

$$F_r = \frac{V}{\sqrt{g * L}} \quad (1)$$

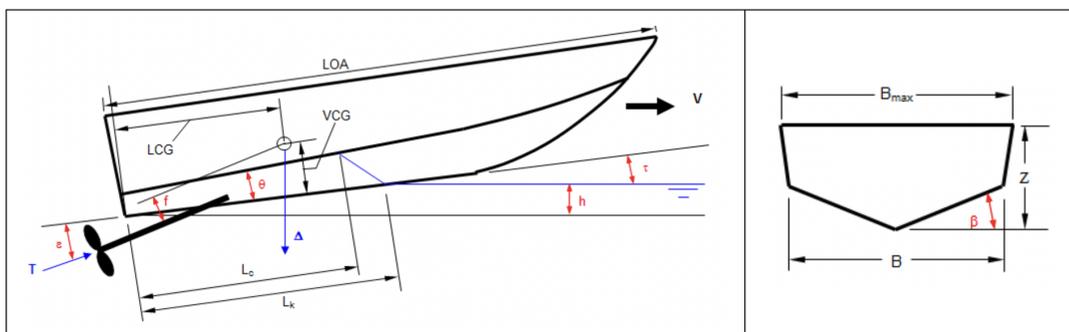
Onde:

V - Velocidade;

L - Comprimento.

O comportamento hidrodinâmico de uma embarcação planante depende fortemente da geometria de seu casco, cujas dimensões e ângulos principais podem ser visualizados na Figura 6. Tais parâmetros ainda serão discutidos na seção Método Savitsky e suas definições encontram-se na lista de símbolos deste trabalho.

Figura 6 – Principais dimensões de uma embarcação planante (vista lateral à esquerda) e ângulo de deadrise (vista frontal à direita)



Fonte: Tweedie (2011)

Entre os parâmetros que mais influenciam o comportamento hidrodinâmico de embarcações planantes, destaca-se o ângulo de *deadrise* β , o qual, segundo Dias Neto, Onety e Tancredi (2021), impacta tanto a resistência ao avanço quanto a aceleração vertical da embarcação.

Nesse sentido, Marin (2015) afirma que o ângulo de *deadrise* é utilizado para caracterizar as regiões nas quais a lancha foi desenvolvida para navegar. Em outras palavras, valores de β inferiores a 17° destinam-se a águas abrigadas, enquanto valores acima de 17° a águas abertas. No entanto, buscando projetos mais eficientes, é comum que o ângulo de *deadrise* varie ao longo do comprimento da embarcação, apresentando valores menores na popa e maiores na proa.

Já Prabowo et al. (2022) explicam que uma mudança no ângulo de *deadrise* provoca a alteração do ângulo de trim que, por sua vez, afeta a estabilidade da embarcação, pois, a baixas velocidades, ângulos de trim elevados podem interferir na estabilidade transversal do navio.

É por essas razões que o ângulo de *deadrise* é um dos mais importantes parâmetros utilizados no modelo proposto por Savitsky (1964) para estimar a resistência ao avanço de um casco planante.

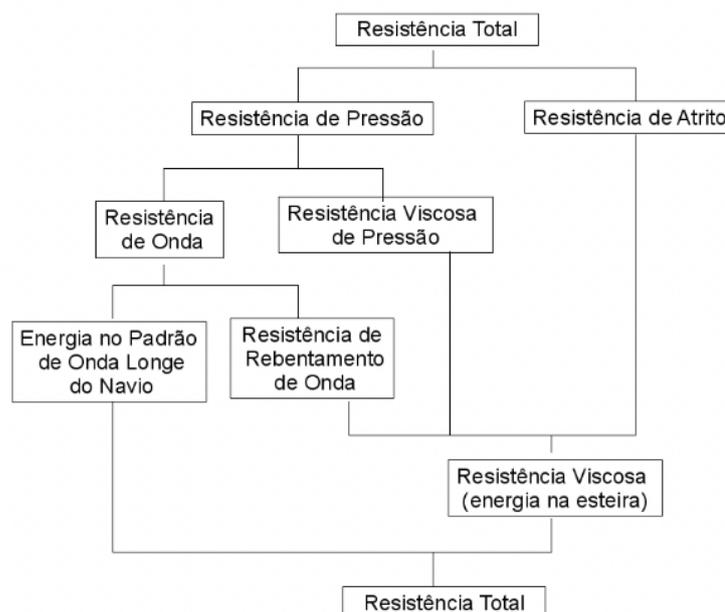
2.3 RESISTÊNCIA AO AVANÇO

De acordo com Bertram (2000), a análise hidrodinâmica de uma embarcação pode ser dividida em: propulsão, comportamento no mar e manobrabilidade. Em virtude do escopo deste trabalho, nesta seção são apresentados o conceito geral e os diferentes métodos disponíveis para estimar a resistência ao avanço de uma embarcação.

2.3.1 Conceito geral

A definição de resistência ao avanço diz respeito à força necessária para rebocar um navio a uma velocidade constante, considerando a navegação em águas tranquilas. Com uma origem complexa, essa força pode, em se tratando de embarcações deslocantes, ser decomposta nas parcelas mostradas na Figura 7 (TRINDADE, 2012).

Figura 7 – Componentes da resistência ao avanço



Fonte: Trindade (2012, p. 19).

Embora possam existir diferentes agrupamentos, em essência, a decomposição da resistência total pode ser descrita pela Equação 2, onde a resistência de atrito refere-se às forças tangenciais que o fluido exerce no casco, enquanto que as demais parcelas referem-se à alteração do campo de pressão no entorno do casco (PRABOWO et al., 2022).

$$R_T = R_F + R_V + R_W \quad (2)$$

Onde:

R_T - Resistência total;

R_F - Resistência de atrito;

R_V - Resistência de pressão viscosa;

R_W - Resistência de onda.

Para facilitar a análise dos fenômenos, usualmente, as componentes da resistência ao avanço também são expressas por meio de coeficientes adimensionais (Equação 3). Essa abordagem facilita a comparação de navios com portes e características distintas (BERTRAM, 2000).

$$c_i = \frac{R_i}{(1/2) * \rho * V^2 * S} \quad (3)$$

Onde:

c_i - Coeficiente de resistência adimensional;

R_i - Resistência i ;

ρ - Massa específica da água;

V - Velocidade do navio;

S - Área molhada do casco.

Trindade (2012) afirma ainda que outras componentes adicionais também devem ser consideradas no cálculo de resistência ao avanço, tais como: resistência aerodinâmica, resistência adicional causada pelas ondas do mar e resistência devida a presença de apêndices no casco ¹.

Ainda de acordo com Trindade (2012), a resistência de atrito é resultado da viscosidade do fluido, cuja interação com a embarcação dependerá do número de Reynolds (R_n), bem como da rugosidade da superfície do casco. Entre os diferentes modelos disponíveis na literatura destinados à estimativa da resistência de atrito de

¹ A favor da concisão, omite-se neste texto a discussão sobre essas parcelas de menor importância. No entanto, o leitor interessado poderá encontrar farta documentação, por exemplo, em Bertram (2000).

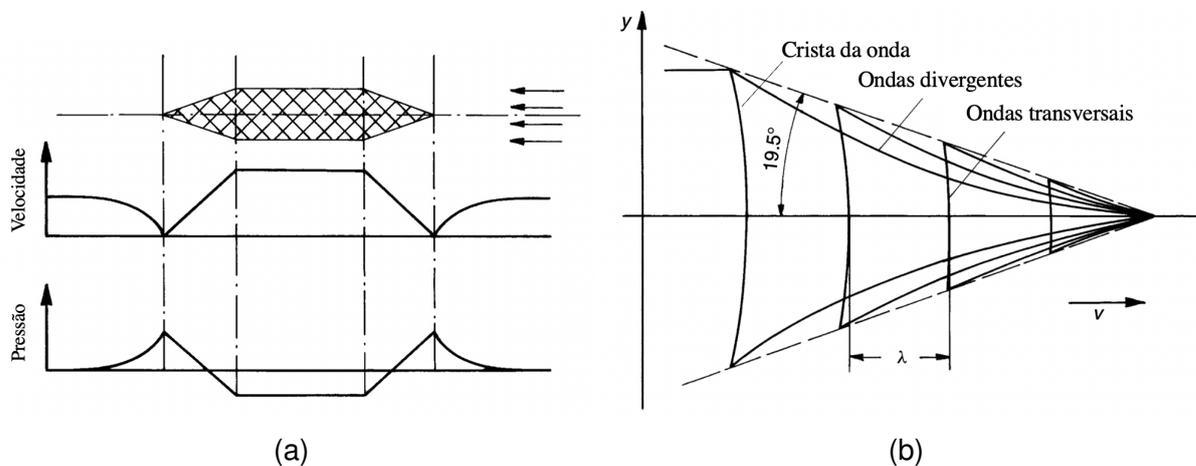
uma embarcação, destaca-se a formulação vista na Equação 4, a qual é proposta por ITTC (1957).

$$c_F = \frac{0,075}{(\log R_n - 2)^2} \quad (4)$$

Um corpo totalmente submerso não apresentará resistência de onda, porém, a resistência total será superior ao valor da resistência de atrito. Isso ocorre pois a forma do casco induz uma mudança local na velocidade do escoamento, que, devido a viscosidade do fluido, é incapaz de recuperar o campo de pressão na popa do navio, induzindo uma força de arrasto conhecida por resistência de pressão viscosa (BERTRAM, 2000).

Por fim, a resistência de onda, dependente da velocidade e da geometria submersa, sendo o resultado da interação do casco com a superfície da água. Nesse sentido, Bertram (2000) mostra que o padrão de ondas formado a ré de uma embarcação é típico, podendo ser decomposto em um sistema primário (Figura 8a), e secundário (Figura 8b).

Figura 8 – Sistema de onda primário (a) e sistema de onda secundário (b)



Fonte: Adaptado de Bertram (2000, p. 67).

No caso de um fluido ideal, não viscoso, o padrão de onda primário representado na Figura 8a apresenta pontos de máxima pressão na proa e na popa do navio. Assim, segundo Trindade (2012), um navio pode ser considerado como um ponto de pressão em movimento, o qual produz um padrão de ondas composto por ondas divergentes e ondas transversais, os quais, em águas profundas, apresentam um meio ângulo característico de 19,5°, independente da forma do casco.

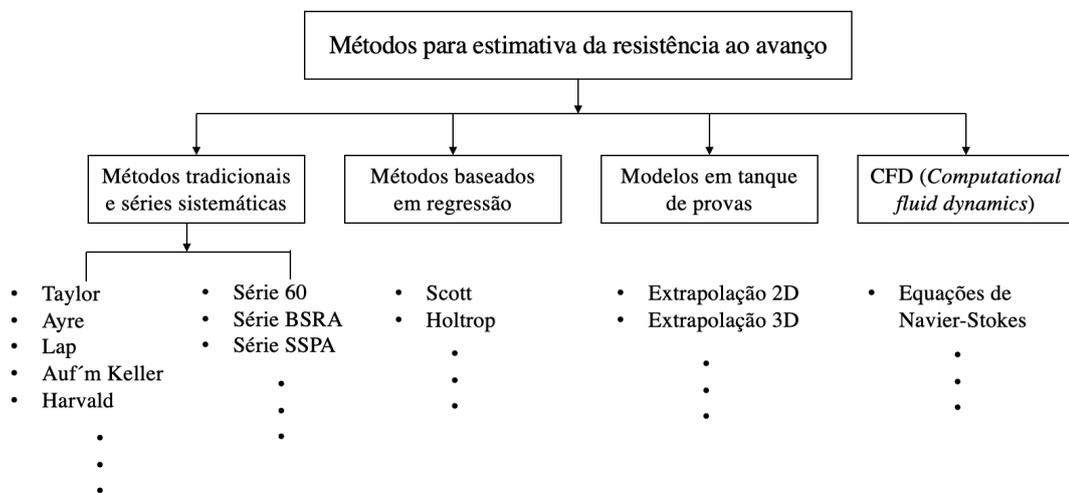
Em geral a resistência de ondas não é facilmente modelada por uma única equação analítica, sendo necessário recorrer a métodos numéricos e/ou experimentais

para determinar essa componente da resistência total (BERTRAM, 2000).

2.3.2 Métodos para estimar a resistência ao avanço

Tal como pode ser visto na Figura 9, existem diversas opções para estimar a resistência ao avanço de uma embarcação e a escolha é, usualmente, baseada em fatores como a precisão desejada, informações disponíveis e até mesmo no recurso financeiro à disposição (MOLLAND, 2008).

Figura 9 – Métodos e exemplos para estimativa da resistência ao avanço



Fonte: Adaptado de Molland (2008, p. 195).

Diferentemente da abordagem CFD e dos testes em tanque de provas, os demais métodos buscam estimar a resistência ao avanço de uma embarcação considerando apenas os principais parâmetros referentes a forma do casco e a distribuição de massa a bordo.

Devido principalmente à simplicidade e ao baixo custo computacional, os métodos empíricos e as séries sistemáticas ainda hoje são bastantes usadas por projetistas navais. Entretanto, com o aumento da disponibilidade de recursos computacionais e a busca cada vez mais intensa por diferenciais competitivos de mercado, projetistas navais tem explorado soluções com geometrias mais complexas, para as quais as abordagens tradicionais podem não ser eficazes na estimativa da resistência ao avanço da embarcação (MOLLAND, 2008).

É importante destacar que uma das abordagens mais utilizadas em projetos navais é a formulação empírica proposta por Holtrop e Mennen (1982), que consiste em um método de regressão estatística para a estimativa, ainda nas fases iniciais de projeto, da resistência ao avanço requerida de uma embarcação².

² O leitor interessado poderá encontrar mais informações, incluindo os limites de validade desse método, em Holtrop e Mennen (1982).

Assim como o método de Holtrop e Mennen (1982), muitas das técnicas descritas na Figura 9 são aplicáveis somente a embarcações deslocantes. Porém, para o projeto de embarcações planantes outros métodos e séries podem ser utilizados. Nesse sentido destaca-se o modelo desenvolvido por Savitsky (1964), amplamente reconhecido na literatura (PRABOWO et al., 2022) (KUKNER; YASA, 2011) por estimar adequadamente a resistência ao avanço de cascos planantes.

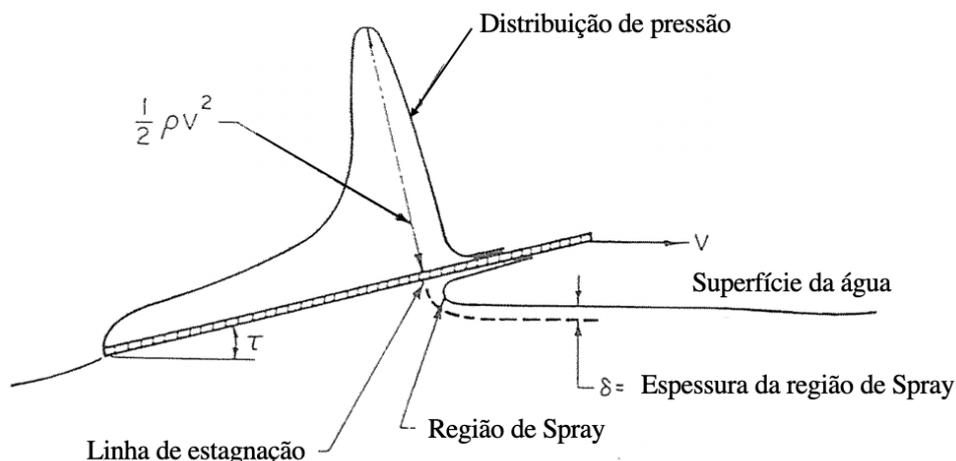
2.3.2.1 Método Savitsky

Dentre os modelos empíricos disponíveis para a estimativa da resistência ao avanço em embarcações planantes, talvez o mais popular seja o proposto por Savitsky (1964). Seu trabalho é até hoje considerado um verdadeiro marco no projeto de cascos planantes e amplamente referenciado na literatura especializada (RIBEIRO, 2002).

Savitsky (1964) discute as principais características hidrodinâmicas de superfícies planantes prismáticas e propõe equações empíricas que descrevem a sustentação, o arrasto (*drag*), a área molhada e o centro de pressão, bem como estabelece os limites de estabilidade dinâmica longitudinal (*porpoising*) em função da velocidade, ângulo de trim, ângulo de *deadrise* e distribuição de massa a bordo. Tais resultados foram utilizados pelo autor a fim de formular um procedimento simples para estimar o desempenho hidrodinâmico de cascos planantes.

Savitsky (1964) inicia sua análise descrevendo a formação de ondas em uma placa plana sem ângulo de *deadrise*, mapeando a distribuição de pressão existente na superfície molhada da placa (Figura 10). Nesse sentido, o autor define o conceito de *wetted area* como sendo a parcela da área sobre a qual pressão é exercida e que exclui a região de spray.

Figura 10 – Distribuição de pressão em uma placa plana



Fonte: Adaptado de Savitsky (1964, p. 72).

De acordo com Savitsky (1964), a sustentação em uma placa plana pode

ser atribuída a dois efeitos distintos, o primeiro diz respeito à reação dinâmica do fluido contra a superfície em movimento, já o segundo relaciona-se à contribuição da flutuação, causada pela pressão estática. Assim, no caso de velocidades baixas, a componente estática predomina, sendo que à medida que a velocidade aumenta, os efeitos dinâmicos de sustentação passam a se sobrepor aos efeitos hidrostáticos.

Alguns dos coeficientes e parâmetros utilizados nas análises de Savitsky (1964) são enumerados no Quadro 1 e foram baseados nas leis de Froude de similaridade. É importante observar que, diferentemente de navios deslocantes, o autor afirma que a boca é a principal dimensão para adimensionalizar os coeficientes de embarcações planantes, uma vez que o comprimento molhado do casco pode variar significativamente com a mudança do trim, do carregamento e/ou da velocidade.

Quadro 1 – Coeficientes e parâmetros presentes no modelo de Savitsky

Símbolo	Descrição	Equação
C_F	Coeficiente de atrito turbulento de Schoenherr	$C_f = D_f \left(\frac{2 \cos \beta}{\rho V_1^2 \lambda b^2} \right)$
C_{L0}	Coeficiente de sustentação para $\beta = 0$	$C_{L0} = \frac{\Delta}{0,5 * \rho * V^2 * b^2}$
$C_{L\beta}$	Coeficiente de sustentação para $\beta \neq 0$	$C_{L\beta} = \frac{\Delta}{0,5 * \rho * V^2 * b^2}$
C_v	Coeficiente de velocidade	$C_v = \frac{V}{(g * b)^{0,5}}$
C_p	Distância do centro de pressão da popa transom	$C_p = \frac{l_p}{\lambda * b}$
Re	Número de Reynolds para superfícies planantes	$Re = \frac{V_1 * \lambda * b}{\nu}$

Fonte: Savitsky (1964, p. 71).

Após tratar de placas planas, Savitsky (1964) desenvolve suas análises para superfícies com ângulo de *deadrise* e que experienciam os mesmos efeitos hidrodinâmicos de sustentação descritos anteriormente. Nesse sentido, a resistência hidrodinâmica total (Equação 5) neste tipo de superfície pode ser decomposta na resistência de pressão D_p , causada pela pressão normal ao fundo inclinado, e pela resistência viscosa D_f , que atua tangencialmente na área molhada e na região de spray.

$$D = \Delta * \tan \tau + \frac{\rho * V_1^2 * C_f * \lambda * b^2}{2 * \cos \beta * \cos \tau} \quad (5)$$

Onde:

Δ - Deslocamento estático;

τ - Ângulo de *trim*;

- ρ - Densidade do fluido;
- V_1 - Velocidade média no fundo da placa;
- C_f - Coeficiente de atrito turbulento de Schoenherr;
- λ - Razão comprimento-boca molhada;
- b - Boca;
- β - Ângulo de *deadrise*;

Nota-se que o termo λb^2 representa a área molhada no fundo da placa e que sua utilização é recomendada para ângulos de trim inferiores a 4° . Além disso, o termo V_1 diz respeito a superfícies sem ângulo de *deadrise*, porém Savitsky (1964) estabelece instruções para a formulação deste termo para os casos em que o ângulo de *deadrise* é diferente de zero (Equação 6).

$$V_1 = V \left(1 - \frac{C_{L\beta}}{\lambda \cos \tau} \right) \quad (6)$$

Por fim, com auxílio dos coeficientes descritos no Quadro 1, Savitsky propõe equações empíricas destinadas à estimar a sustentação em placas planas com e sem ângulo de *deadrise*, cujos detalhes podem ser vistos em Savitsky (1964).

Destaca-se ainda que o modelo de Savitsky (1964) adequa-se a cascos prismáticos. Nesse sentido, para formas de casco com ângulo de *deadrise* e distribuição de boca variáveis ao longo do comprimento, faz-se necessário introduzir correções ao modelo, tal como discutido por Ribeiro (2002).

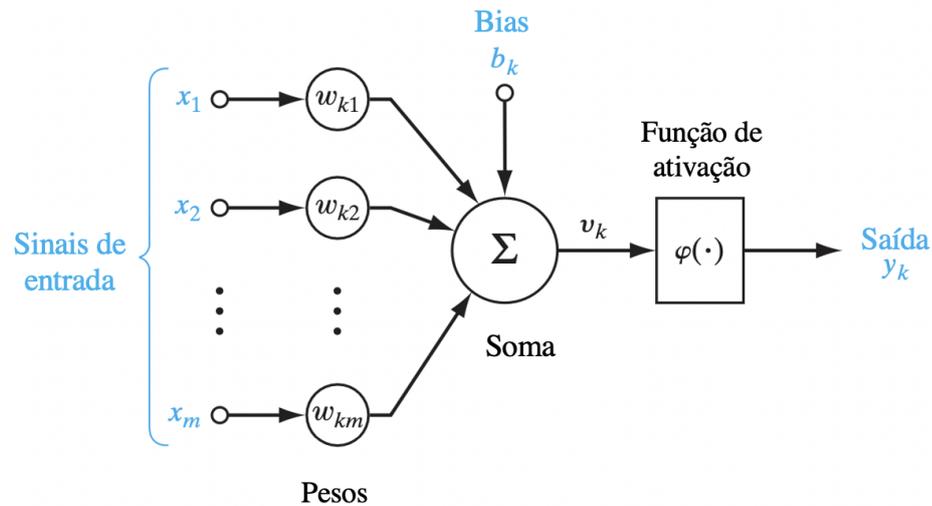
2.4 REDES NEURAIS

Uma rede neural artificial é um sistema de processamento de informações que possui características semelhantes àsquelas observadas em redes neurais biológicas, uma vez que apresenta generalizações de modelos matemáticos referentes à cognição humana. Nesse sentido, a caracterização de uma rede neural é dada por sua topologia, treinamento (ou algoritmo de aprendizado) e função de ativação (FAUSETT, 1993).

2.4.1 Unidade neural

O primeiro modelo matemático que emula o comportamento funcional de um neurônio foi concebido por Mcculloch e Pitts (1943). Neste modelo, aspectos biológicos fundamentais como o recebimento de impulsos elétricos, estímulo interno e transmissão final são inseridos em um cenário lógico por meio de vetores de entrada e saída, pesos e valores, cujo funcionamento dependerá de uma função denominada de função de ativação (Figura 11).

Figura 11 – Modelo não linear de um neurônio



Fonte: Adaptado de Haykin (2008, p. 11).

Neste modelo, o neurônio representado pelo índice k receberá uma entrada vetorial com valores x_j , os quais serão multiplicados por seus respectivos pesos w_j e, finalmente, acumulados por meio de um somatório. O valor da saída, dado por y_k , será o resultado da função de ativação φ estabelecida para o neurônio, conforme a Equação 7, descrita, por exemplo, em Haykin (2008).

$$y_k = \varphi \left(\sum_{j=1}^m w_{kj} * x_j + b_k \right) \quad (7)$$

O modelo representado na Figura 11 apresenta ainda um valor bias b_k que é acrescentado visando aumentar ou diminuir o valor de entrada da função de ativação, permitindo deslocar o valor de saída para se adequar aos dados do problema (HAYKIN, 2008).

Por fim é necessário definir também um valor conhecido como limitador t ou do inglês *threshold value*, aplicado a cada neurônio e responsável por delimitar o valor mínimo responsável pela ativação do neurônio. A definição desse parâmetro é comumente incluída na definição da função de ativação utilizada no modelo (KRIESEL, 2007).

2.4.2 Funções de ativação

Como visto anteriormente, a operação básica de um neurônio artificial envolve a saída de um valor escalar y_k , o qual é calculado utilizando uma função de ativação. Usualmente essa função é não linear e é aplicada igualmente em todos os neurônios que formam a rede neural (FAUSETT, 1993).

Segundo Juan e Valdecantos (2022), os principais tipos de funções de ativação utilizadas em problemas de engenharia naval e oceânica são:

- Função linear;
- Função sigmoide;
- Função tangente hiperbólica;
- Função base radial.

Uma vez que a escolha da função de ativação representa uma parte crítica da concepção de uma rede neural, torna-se necessário conhecer a fundo as aplicabilidades, vantagens e desvantagens de cada uma com relação ao tipo de problema a ser modelado. No entanto, descrever todas as nuances pertinentes a essa escolha fugiria ao escopo deste trabalho. Diante disso, são apresentadas no Quadro 2 somente o equacionamento e as principais aplicações de cada uma das funções citadas anteriormente.

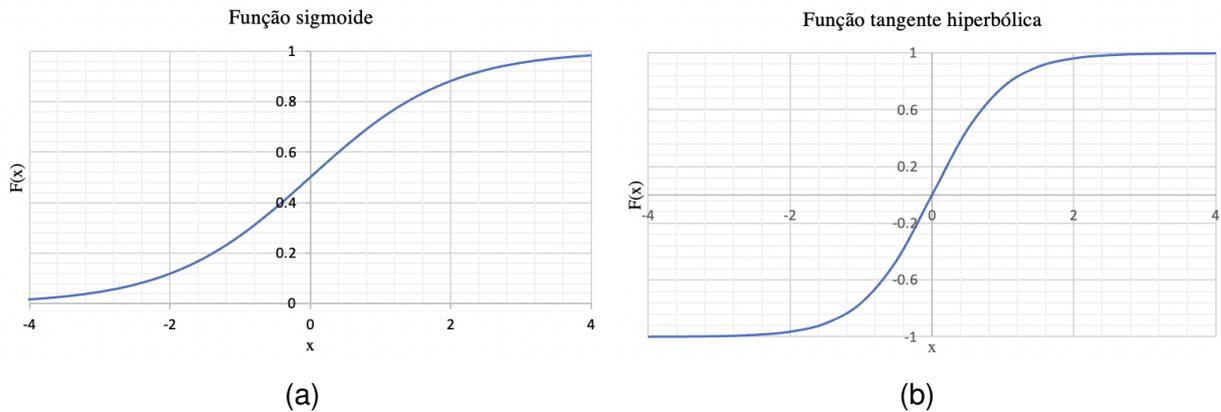
Quadro 2 – Funções de ativação mais comuns utilizadas em problemas de engenharia naval e oceânica

Função	Expressão	Aplicações
Linear	$F(x) = x$	Camadas de saída.
Sigmoide	$F(x) = \frac{1}{1+\exp(-x)}$	Camadas intermediárias e de saída. Intervalo [0,1].
Tangente Hiperbólica	$F(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$	Camadas intermediárias. Intervalo [-1,1].
Base radial	$F(x) = \exp\left(-\frac{(x-\mu)^2}{\sigma^2}\right)$	Camadas intermediárias e de saída.

Fonte: Adaptado de Juan e Valdecantos (2022, p. 2).

Tanto o comportamento da função sigmoide quanto o da função tangente hiperbólica podem ser visualizados na Figura 12. Nota-se ainda que ambas as funções são diferenciáveis em toda a sua extensão, fato que viabiliza a implementação do processo de aprendizado denominado *Backpropagation* (KRIESEL, 2007).

Figura 12 – Comportamento da função sigmoide (a) e função tangente hiperbólica (b)



Fonte: Autora (2022).

2.4.3 Topologia da rede neural

Após entender o funcionamento básico da unidade neural, pode-se definir o conceito de rede neural e, conseqüentemente, topologia. Neste sentido, uma rede neural é um processador distribuído paralelamente constituído de unidades de processamento (neurônios), que apresentam uma habilidade natural para armazenar conhecimento e reconhecer padrões (HAYKIN, 2008).

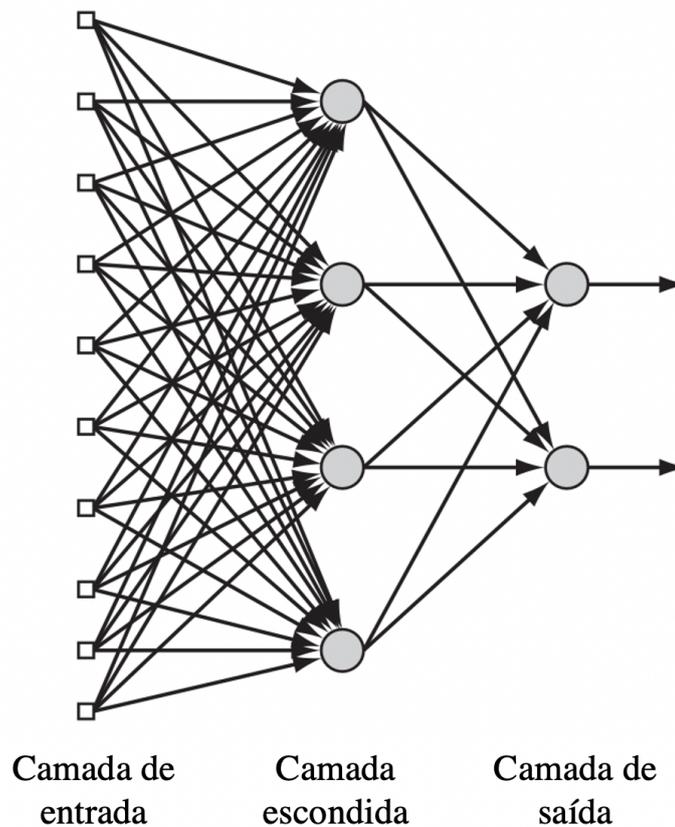
Por topologia compreende-se o arranjo, ou seja, a arquitetura, com a qual a rede é estruturada. Segundo Haykin (2008) há três formas usuais de se estruturar uma rede neural:

- *Feedforward* com camada simples;
- *Feedforward* com camadas múltiplas;
- Recorrente.

A diferença entre as redes de camada única e as redes de múltiplas camadas está no número de camadas internas (escondidas) presentes na estrutura. Ao inserir camadas intermediárias, a rede pode, potencialmente, ser capaz de identificar padrões de maior complexidade (FAUSETT, 1993).

Além da quantidade de camadas internas, a rede também pode ser definida pela direção na qual o fluxo de sinais percorre os neurônios. As redes nas quais esses sinais partem da camada de entrada e seguem em direção à camada de saída são denominadas *Feedforward* (FAUSETT, 1993). Para ilustrar essa discussão, na Figura 13 é mostrado um exemplo desse tipo de arquitetura, na qual é possível observar uma camada interna e uma camada de saída.

Figura 13 – Rede neural *Feedforward* com múltiplas camadas



Fonte: Adaptado de Haykin (2008, p. 11).

Por último, redes com estruturas recorrentes se distinguem das redes *Feedforward* por apresentarem no mínimo um loop no fluxo de sinais (FAUSETT, 1993). Em outras palavras, os sinais de saída provenientes de neurônios de uma camada são utilizados como entrada para unidades da mesma camada ou de outras.

Em geral redes *Feedforward* são utilizadas para a construção de modelos preditivos, enquanto redes com estrutura recorrente são utilizadas no reconhecimento de padrões.

2.4.4 Aprendizado e treinamento da rede neural

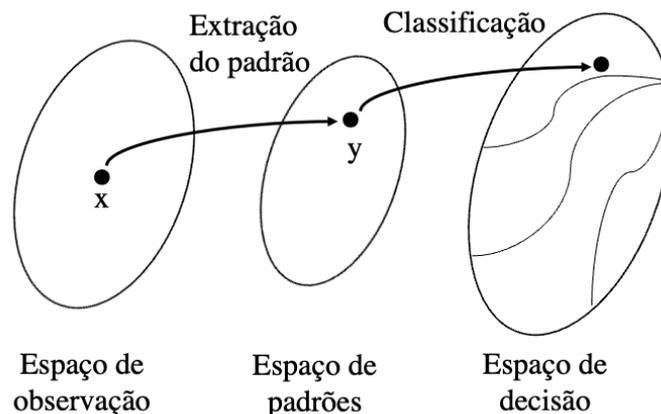
Conforme visto em Kriesel (2007), uma característica importante das redes neurais é a chamada generalização, ou seja, a capacidade de se familiarizar com determinados problemas por meio de treinamento e, por conseguinte, serem capazes de resolver novas situações utilizando os padrões identificados a partir de um histórico conhecido.

O processo pelo qual uma rede neural identifica padrões a partir de um histórico conhecido é denominado de aprendizado ou treinamento. Sobre esse assunto, Kriesel (2007) destaca ainda que os algoritmos por trás das redes neurais podem funcionar de diferentes formas. Assim, a depender do processo de aprendizado utilizado, é

possível deletar ou criar conexões; alterar os pesos de conexão ou o limite de ativação e até mesmo criar ou excluir neurônios. Todavia, o procedimento de treinamento mais comum consiste na alteração dos pesos w_j e dos limites de ativação dos neurônios pertencentes à rede neural.

Tal como os humanos podem aprender de diferentes maneiras, as redes neurais também o fazem. Nesse contexto, os processos de aprendizado existentes são definidos e diferenciados em relação ao seu respectivo conjunto de treinamento (KRIESEL, 2007). Assim, de acordo com Haykin (2008), o primeiro processo diz respeito ao aprendizado não supervisionado, no qual a rede recebe como entrada um conjunto de treinamento e tenta, por si mesma, detectar similaridades entre os dados apresentados, buscando identificar os padrões existentes (Figura 14).

Figura 14 – Abordagem para a classificação de padrões



Fonte: Adaptado de Haykin (2008, p. 11).

O segundo processo de aprendizado é chamado de supervisionado, sendo que nesse caso, o conjunto de treinamento apresenta respostas desejáveis para o problema em questão implicando na comparação direta entre os valores reais e estimados. Por último, tem-se o aprendizado com reforço, que assim como o supervisionado também recebe um conjunto de treinamento com padrões, porém, nesse caso, a cada etapa de processamento um valor é retornado à rede a fim de indicar quão próximas as estimativas realizadas estão dos resultados conhecidos (HAYKIN, 2008).

Novamente é importante destacar que a chance de uma rede neural identificar padrões dependerá fortemente da quantidade e da qualidade dos dados disponíveis (YANG et al., 2021). Tais dados são comumente divididos em: conjunto de treinamento, conjunto de validação e conjunto de teste. Entretanto, a forma de dividir esses dados nem sempre é a mesma, haja vista a diversidade de problemas e cenários possíveis. Uma primeira opção de divisão consiste na abordagem utilizada por Couser et al. (2004), na qual tem-se uma proporção de 70%, 15% e 15%, enquanto Abramowski (2013) utiliza uma proporção de 70%, 20% e 10%.

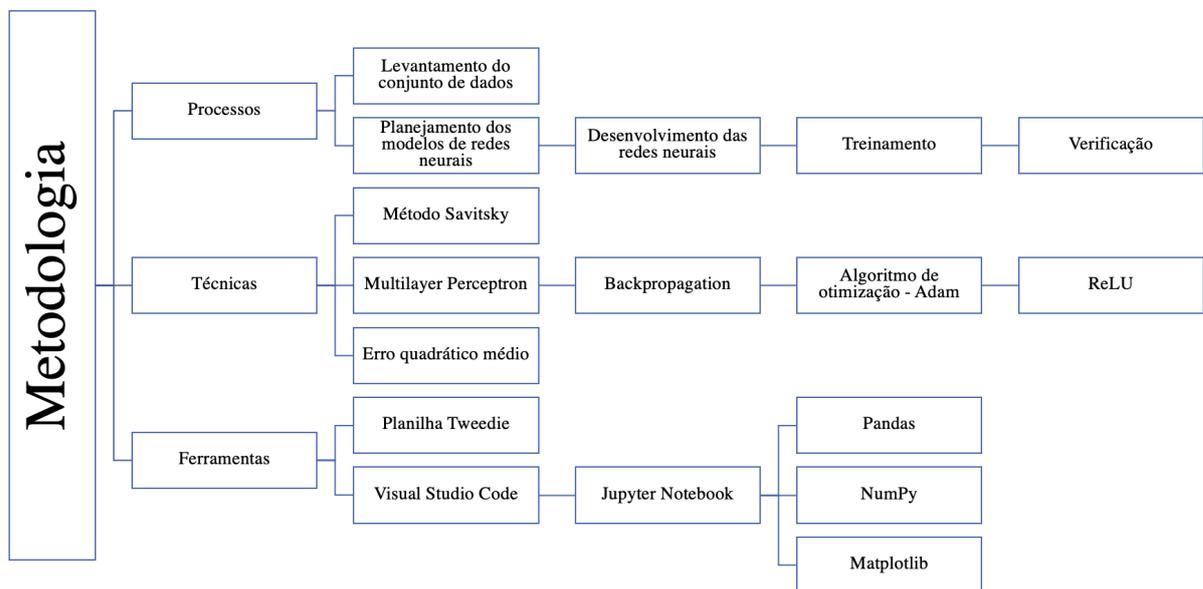
A precisão obtida pela rede neural desenvolvida é, por fim, calculada por meio de uma abordagem estatística (CEPOWSKI, 2020). Tendo em vista a pluralidade de abordagens existentes, é comum a verificação da rede usando o erro quadrático médio (EQM) calculado por meio da Equação 8, onde n representa o número total de dados analisados pela rede, seja na fase de treinamento ou de teste.

$$EQM = \frac{\sum (x_{\text{atual}} - x_{\text{estimado}})^2}{n} \quad (8)$$

3 METODOLOGIA

Apresenta-se neste capítulo a metodologia empregada na execução deste trabalho, a qual é sintetizada na Figura 15. Nesse sentido, são apontados os processos, técnicas e ferramentas pertinentes ao desenvolvimento, treinamento e verificação da rede neural desenvolvida para estimar a resistência ao avanço de embarcações planantes.

Figura 15 – Metodologia empregada



Fonte: Autora (2022).

Inicialmente foi realizado o levantamento dos dados, buscando identificar embarcações planantes cujas informações referentes ao deslocamento, a boca, ao comprimento, ao ângulo de *deadrise* na popa transom e na meia nau estivessem disponíveis na literatura.

A seguir, foi calculada a curva de resistência ao avanço para cada uma das embarcações identificadas na fase de levantamento de dados, sendo que essa análise foi realizada utilizando-se o método de Savitsky (1964), cuja implementação foi feita por Tweedie (2011) e discutida por Allebrandt Junior (2021) e Marin (2015).

Os parâmetros referentes às embarcações planantes enumeradas na fase de levantamento de dados, juntamente com os valores de resistência ao avanço calculados utilizando o método de Savitsky (1964) implementado por Tweedie (2011) formam o conjunto de dados utilizados neste trabalho.

Dado que um dos objetivos estabelecidos para este trabalho foi avaliar o impacto da topologia e dos parâmetros de entrada usados no treinamento das redes

neurais, foram elaborados 12 estudos de caso, cujas informações são sistematizadas na Tabela 1.

Tabela 1 – Modelos de redes neurais propostos

Modelo	Parâmetros de entrada	Nº de neurônios (primeira camada)	Nº de camadas internas	Nº de neurônios (camada interna)	Nº de neurônios (última camada)	Tamanho
M1	G1	6	1	3	1	06x3x1
M2	G1	6	1	5	1	06x5x1
M3	G1	6	2	3	1	06x3x1
M4	G1	6	2	5	1	06x5x1
M5	G2	9	1	5	1	09x5x1
M6	G2	9	1	7	1	09x7x1
M7	G2	9	2	5	1	09x5x1
M8	G2	9	2	7	1	09x7x1
M9	G3	13	1	7	1	13x7x1
M10	G3	13	1	9	1	13x9x1
M11	G3	13	2	7	1	13x7x1
M12	G3	13	2	9	1	13x9x1

Fonte: Autora (2022).

É importante mencionar que o planejamento realizado teve como inspiração os modelos vistos em Yang et al. (2021), onde a nomenclatura G1, G2 e G3 descreve o grupo de parâmetros navais utilizados em cada uma das redes estudadas, tal como pode ser visto na Tabela 2.

Tabela 2 – Grupos de parâmetros navais de entrada

Grupo	Parâmetros do grupo
G1	$\Delta_{\text{Total}}, LWL, \beta_T, \beta_x, B_{\text{max}}, V$
G2	$\Delta_{\text{Total}}, LWL, \beta_T, \beta_x, B_{\text{max}}, V, L_x, LCG, VCG$
G3	$\Delta_{\text{Total}}, LWL, \beta_T, \beta_x, B_{\text{max}}, V, L_x, LCG, VCG, Z, LOA, LWL/B, T_{\text{max}}$

Fonte: Autora (2022).

A seleção dos parâmetros navais presentes nos grupos G1, G2 e G3 (parâmetros de entrada) foi baseada em trabalhos como Yang et al. (2021), Dias Neto, Onety e Tancredi (2021) e Abramowski (2013), tendo em vista a relevância desses parâmetros na estimativa da resistência ao avanço de embarcações planantes.

Já para o número de camadas internas foram utilizadas no máximo duas, pois, segundo Tancredi (2008), redes neurais com até duas camadas internas são capazes de determinarem padrões na maioria dos problemas não caóticos, como, por exemplo, a estimativa da resistência ao avanço de uma embarcação (NÉIA, 2016).

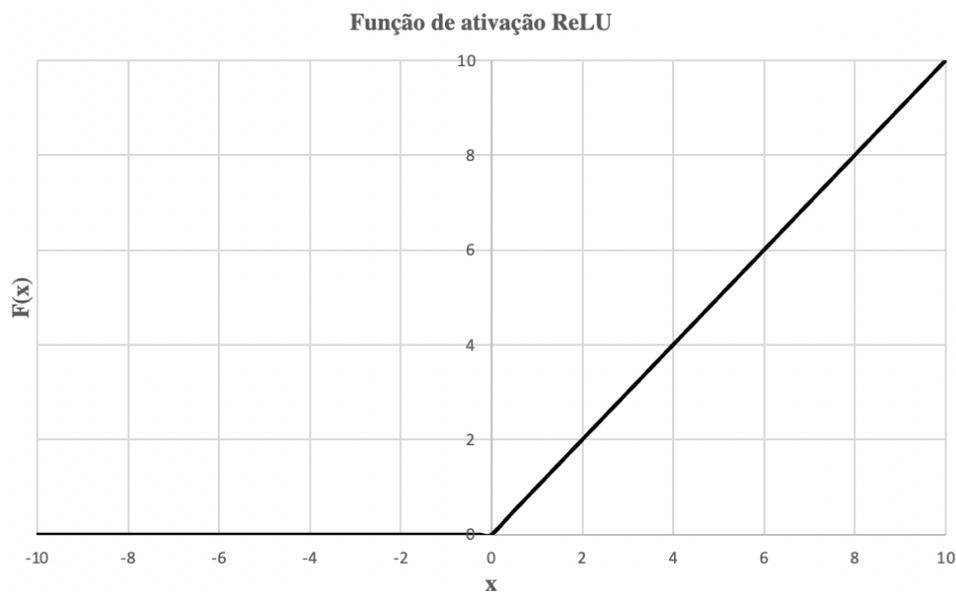
Por fim, seguindo a recomendação de Heaton (2017), o número de neurônios nas camadas internas foi variado de sorte a sempre estar no intervalo descrito pelo número de neurônios na camada de entrada e o número de neurônios na camada de

saída.

Em virtude do tipo de problema modelado, todas as topologias de redes neurais estudadas apresentam um único neurônio na camada de saída, o qual corresponde à resistência ao avanço calculada em função dos parâmetro de entrada do modelo, os quais incluem a velocidade de avanço da embarcação. Além disso, todas as topologias propostas utilizaram a função de ativação ReLU (*Rectified Linear Unit*) nas camadas internas e a função linear na camada de saída.

A função de ativação ReLU, mostrada na Figura 16, foi escolhida para as camadas internas, em especial, por apresentar uma performance computacional melhor em termos de velocidade quando comparada a outros tipos de funções complexas de ativação. A rápida convergência apresentada pela função ReLU ocorre pela inexistência de termos exponenciais, além da existência de uma derivada fixa para valores de X positivos e uma derivada nula para valores de X negativos, características que aceleram o processo de aprendizado da rede (PRAMODITHA, 2019).

Figura 16 – Função de ativação ReLU



Fonte: Autora (2022).

Conforme já mencionado, o desenvolvido considerou a utilização de redes com múltiplas camadas, as quais também são chamadas de redes MLP (*Multilayer Perceptron*), sendo que o algoritmo de treinamento utilizado foi *Backpropagation* realizado com o auxílio do algoritmo Adam Optimizer, cujos códigos foram implementados em Python, versão 3.10, por meio do editor de código Visual Studio Code e do ambiente interativo de desenvolvimento Jupyter Notebook. Além disso, para facilitar a manipulação e análise dos dados, foram utilizadas as bibliotecas Pandas, NumPy e Matplotlib.

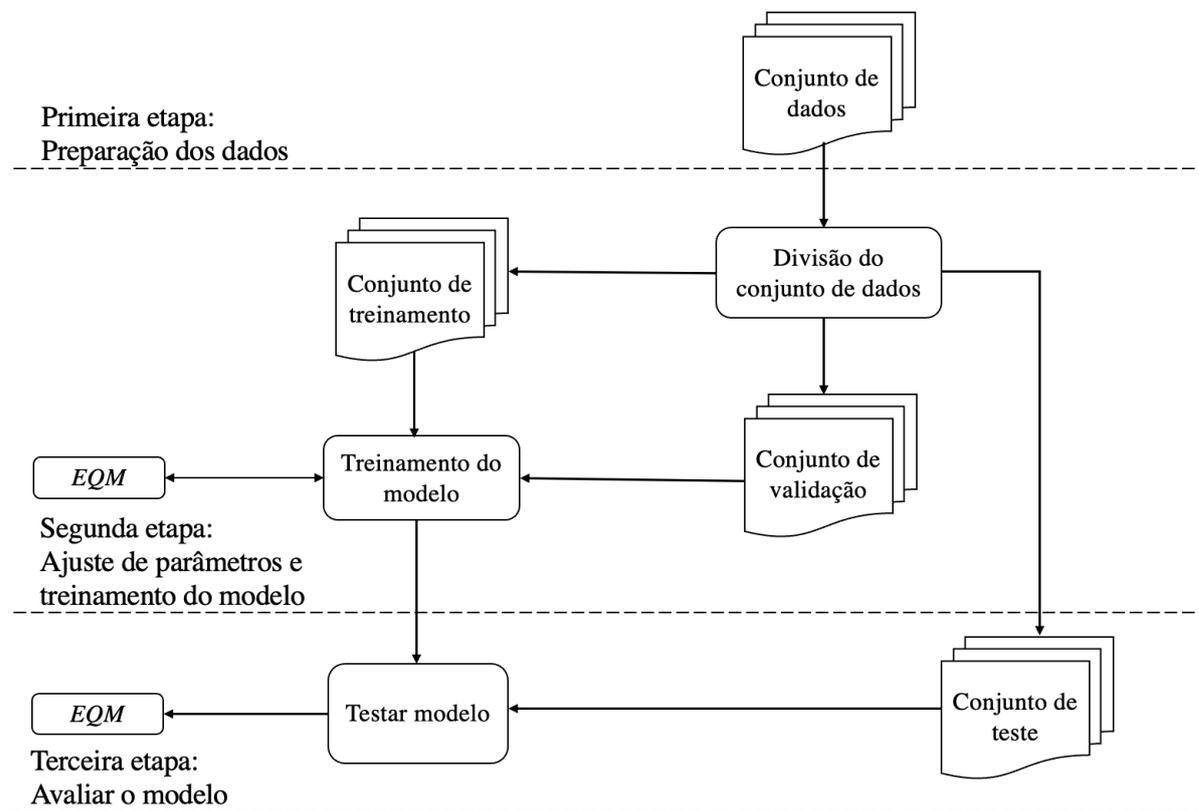
A escolha da linguagem de programação Python baseou-se no amplo

histórico de utilização dessa linguagem em aplicações de *Machine Learning*, além da sintaxe simples e de fácil aprendizado. Embora seja uma linguagem interpretada, o Python possui uma excelente performance, pois consegue acessar as principais bibliotecas, escritas em linguagem de alto desempenho como Fortran, por meio de encapsulamentos.

O treinamento de cada uma das redes neurais estudadas seguiu o fluxograma mostrado na Figura 17, aplicando-se o erro quadrático médio tanto para o ajuste dos pesos da rede, quanto para quantificar o desempenho do modelo durante a fase de avaliação.

Seguindo o que foi discutido na fundamentação teórica, optou-se pela divisão do conjunto de dados seguindo a proporção de 70% para o conjunto de treinamento, 15% para o conjunto de validação e 15% para o conjunto de teste.

Figura 17 – Fluxograma de treinamento e avaliação das redes neurais



Fonte: Autora (2022).

Por fim, cada uma das redes estudadas teve o desempenho medido, permitindo estabelecer recomendações a cerca da melhor combinação entre os grupos de parâmetros de entrada e as topologias das redes neurais destinadas a estimar a resistência ao avanço de uma embarcação planante.

4 DESENVOLVIMENTO

Descreve-se neste capítulo cada uma das etapas do desenvolvimento, passando pelo levantamento de dados, pela implementação dos algoritmos computacionais, pelo treinamento e, por fim, discutindo a verificação dos modelos de redes neurais estudados.

4.1 CONJUNTO DE DADOS

O levantamento de dados buscou identificar embarcações planantes que apresentassem os principais parâmetros de projeto disponíveis na literatura. Para essa busca foram consultados diferentes sites de compra e venda, publicações especializadas sobre o assunto, revistas náuticas e estaleiros. Todos esses dados foram reunidos e dispostos nas tabelas presentes nos Apêndices B, C, D e E, sendo que os dados obtidos nas diversas fontes consultadas foram organizados e sistematizados em uma tabela única, a qual pode ser vista no Apêndice A.

Seguindo a metodologia proposta, finalizado o levantamento de dados, analisou-se cada uma das embarcações selecionadas, calculando-se a resistência ao avanço para sete velocidades (15 , 20, 25, 30, 35, 40 e 45 nós). Os resultados presentes nos Apêndices F e G se referem ao campo $\sum F_{\text{hori}}$ da planilha de Tweedie (2011) e correspondem à resistência ao avanço da embarcação analisada quando navegando em cada uma das velocidades consideradas na análise.

Com o intuito de facilitar a visualização dos dados levantados, foram avaliados os limites inferiores e superiores de cada parâmetro de projeto considerado (Tabela 3), cuja avaliação permite estabelecer os limites e aplicações das redes neurais desenvolvidas.

Tabela 3 – Análise estatística do conjunto de dados utilizado no desenvolvimento

Parâmetro	Unidade	Limite Inferior	Limite Superior	Média	Desvio Padrão
Δ_{Total}	t	1,97	73,70	37,84	50,72
Δ_{Leve}	t	1,72	62,65	32,19	43,08
B	m	2,24	6,29	4,27	2,86
B_{max}	m	2,50	6,38	4,29	2,96
β_T	°	9,50	23,50	16,50	9,90
β_x	°	9,50	23,00	16,25	9,55
T_{max}	m	0,50	1,98	1,24	1,05
Z	m	1,40	6,20	3,80	3,39
LOA	m	7,01	26,84	16,93	14,02
LWL	m	6,03	21,42	13,73	10,88
L_x	m	3,01	10,71	6,86	5,44
LWL/B	-	2,17	3,82	3,00	1,17
B/Z	-	0,73	2,11	1,42	0,98
LCG	m	2,71	9,64	6,18	4,90
VCG	m	0,14	3,10	1,62	2,09
$V_{cruzeiro}$	nós	19,00	40,00	29,50	14,85

Fonte: Autora (2022).

Por fim é importante apontar que, em alguns casos, foi necessário realizar considerações e aproximações visando permitir a aplicação do método de Savitsky destinado a estimar a resistência ao avanço das embarcações analisadas. Entre as hipóteses adotadas, destacam-se:

- $\Delta_{Total} = 1,15 * \Delta_{Leve}$;
- $L_x = LWL/2$;
- $LCG = 0,45 * LWL$;
- $VCG = Z/2$;
- $\theta = 0$;
- $\epsilon = 5$;
- $f = VCG/7$;
- $H_{ss} = 0$;
- $B_{ss} = 0$;
- $A_{ss} = 0$;
- $LWL = 0,86 * LOA$;
- Nos casos em que o pontal moldado não foi fornecido, considerou-se a mesma relação B/Z presente em outros cascos do mesmo fabricante para determinar Z;
- Nos casos em que β_x não foi fornecido, considerou-se que $\beta_x = \beta_T$.

4.2 REDES NEURAIIS

Esta seção tem como objetivo descrever o código elaborado, o qual foi desenvolvido com o auxílio de VanderPlas (2017), Spahic (2020), de fóruns online e de bibliotecas já existentes.

4.2.1 Importação de bibliotecas

A utilização de bibliotecas são uma vantagem em se tratando do conceito de *Open Source*. Elas constituem um conglomerado de códigos que podem ser utilizados diversas vezes em diferentes programas, tornando o processo de programação mais simples e eficiente (SPAHC, 2020).

Para facilitar o desenvolvimento realizado, foram importadas bibliotecas pertinentes a *Machine Learning*. As linhas de código da Figura 18 representam as bibliotecas utilizadas.

Figura 18 – Bibliotecas importadas

```
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from sklearn.model_selection import train_test_split
import numpy as np
import matplotlib.pyplot as plt
from keras_visualizer import visualizer
import datetime
```

Fonte: Autora (2022).

As principais funcionalidades relativas às bibliotecas da Figura 18 podem ser encontradas na Tabela 4. O leitor interessado poderá encontrar mais informações sobre essas e outras bibliotecas disponíveis para Python em VanderPlas (2017)

Tabela 4 – Principais funcionalidades das bibliotecas utilizadas

Biblioteca	Funcionalidade
Pandas	Manipulação de tabelas ou <i>dataframes</i>
Tensorflow	Auxílio para a construção dos modelos de redes neurais
Keras	Desenvolvimento, treinamento e avaliação da rede neural
NumPy	Execução de operações matemáticas em <i>arrays</i> e matrizes
Matplotlib	Visualização dos dados (plotagem de gráficos)
Datetime	Manipulação de datas e horas

Fonte: Autora (2022).

4.2.2 Definição de funções

Já as funções são blocos de códigos executados quando solicitados e que recebem parâmetros de entrada com o objetivo de retornarem dados de saída. Ao longo do desenvolvimento deste trabalho foram definidas diferentes funções, as quais serão discutidas nesta seção.

A primeira função definida é a *readInputParams()* (Figura 19), que tem como objetivo ler os dados referentes às topologias enumeradas na Tabela 1 e que são armazenados no arquivo *params.csv* cuja estrutura é mostrada na Figura 20. A seguir, as informações lidas são convertidas em uma estrutura de dados do tipo *dataframe* pela biblioteca Pandas.

Figura 19 – Função *readInputParams()*

```
def readInputParams(file_name):
    df_params = pd.read_csv(file_name, index_col=False)

    df_params = df_params.rename(columns={
        'modelo': 'name',
        'entrada': 'input_columns',
        'velocidades': 'velocity_columns',
        'camadas internas': 'num_hidden_layers',
        'num neuronios internos': 'num_neurons_per_hidden_layer',
        'funcao de ativacao': 'activation_func'
    })

    df_params['input_columns'] = df_params['input_columns'].apply(lambda value: [x.strip() for x in value.split(',')])
    df_params['velocity_columns'] = df_params['velocity_columns'].apply(lambda value: [x.strip() for x in value.split(',')])

    return df_params.to_dict('records')
```

Fonte: Autora (2022).

Figura 20 – Arquivo *params.csv*

```
modelo,entrada,velocidades,camadas internas,num neuronios internos,funcao de ativacao
m1,"Deslocamento Total, LWL,  $\beta$  (popa= aft),  $\beta x$  (amid), Bmax", "15, 20, 25, 30, 35, 40", 1,3,relu
m2,"Deslocamento Total, LWL,  $\beta$  (popa= aft),  $\beta x$  (amid), Bmax", "15, 20, 25, 30, 35, 40", 1,5,relu
m3,"Deslocamento Total, LWL,  $\beta$  (popa= aft),  $\beta x$  (amid), Bmax", "15, 20, 25, 30, 35, 40", 2,3,relu
m4,"Deslocamento Total, LWL,  $\beta$  (popa= aft),  $\beta x$  (amid), Bmax", "15, 20, 25, 30, 35, 40", 2,5,relu
m5,"Deslocamento Total, LWL,  $\beta$  (popa= aft),  $\beta x$  (amid), Bmax, L_x , LCG, VCG ", "15, 20, 25, 30, 35, 40", 1,5,relu
m6,"Deslocamento Total, LWL,  $\beta$  (popa= aft),  $\beta x$  (amid), Bmax, L_x , LCG, VCG ", "15, 20, 25, 30, 35, 40", 1,7,relu
m7,"Deslocamento Total, LWL,  $\beta$  (popa= aft),  $\beta x$  (amid), Bmax, L_x , LCG, VCG ", "15, 20, 25, 30, 35, 40", 2,5,relu
m8,"Deslocamento Total, LWL,  $\beta$  (popa= aft),  $\beta x$  (amid), Bmax, L_x , LCG, VCG ", "15, 20, 25, 30, 35, 40", 2,7,relu
m9,"Deslocamento Total, LWL,  $\beta$  (popa= aft),  $\beta x$  (amid), Bmax, L_x , LCG, VCG, D, LOA, LWL/B, Tmax[m]", "15, 20, 25, 30, 35, 40", 1,7,relu
m10,"Deslocamento Total, LWL,  $\beta$  (popa= aft),  $\beta x$  (amid), Bmax, L_x , LCG, VCG, D, LOA, LWL/B, Tmax[m]", "15, 20, 25, 30, 35, 40", 1,9,relu
m11,"Deslocamento Total, LWL,  $\beta$  (popa= aft),  $\beta x$  (amid), Bmax, L_x , LCG, VCG, D, LOA, LWL/B, Tmax[m]", "15, 20, 25, 30, 35, 40", 2,7,relu
m12,"Deslocamento Total, LWL,  $\beta$  (popa= aft),  $\beta x$  (amid), Bmax, L_x , LCG, VCG, D, LOA, LWL/B, Tmax[m]", "15, 20, 25, 30, 35, 40", 2,9,relu
```

Fonte: Autora (2022).

Para a preparação dos dados foi desenvolvida a função *preparaDataset()*, a qual recebe dois parâmetros, o arquivo *params.csv* e o conjunto de dados completo denominado *df_original* (Figura 21). Inicialmente essa função filtra do *df_original* somente os dados necessários ao modelo em questão, ou seja, as colunas do arquivo *params.csv* efetivamente utilizadas no treinamento da rede.

A seguir, os dados filtrados são duplicados até que o número de linhas da tabela corresponda a quantidade de velocidades analisadas. Por fim, adiciona-se duas colunas à tabela de dados onde são incluídas as diferentes velocidades analisadas e as

respectivas resistências ao avanço, dando origem ao conjunto de dados denominado `full_df`.

Figura 21 – Função `preparaDataset()`

```
def preparaDataset(params, df_original):

    df = df_original[params['input_columns']]

    duplicated_df = pd.concat([df] * len(params['velocity_columns']), ignore_index=True)

    velo_df = df_original[params['velocity_columns']]

    velo_df = velo_df.melt(var_name='Velocity', value_name='Resistance')

    full_df = duplicated_df.join(velo_df, lsuffix='a', rsuffix='b')

    X = np.asarray(full_df.iloc[:, 0:len(params['input_columns']) + 1].values).astype('float32')
    y = np.asarray(full_df['Resistance'].values).astype('float32')

    return X, y
```

Fonte: Autora (2022).

Conforme pode visto na Figura 21, finalizada a preparação dos dados, a função retorna, o vetor de entrada `X` e o vetor de saída `y`, fundamentais para o treinamento de cada uma das redes neurais estudadas.

Seguindo a metodologia descrita no capítulo 3, o conjunto de dados deste trabalho foi dividido entre conjunto de treinamento, de validação e de teste. Para realizar essa divisão definiu-se a função `splitDataset()`, a qual, usando a biblioteca NumPy, divide o conjunto de dados `full_df` de forma aleatória, mas igualmente replicável a todas as topologias estudadas por meio da variável `random_state` (Figura 22).

Figura 22 – Função `splitDataset()`

```
def splitDataset(X, y, test_size=0.15, val_size=0.15, random_state=42):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=random_state)
    X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=val_size, random_state=random_state)

    X_train = np.asarray(X_train).astype('float32')
    y_train = np.asarray(y_train).astype('float32')
    X_val = np.asarray(X_val).astype('float32')
    y_val = np.asarray(y_val).astype('float32')
    X_test = np.asarray(X_test).astype('float32')
    y_test = np.asarray(y_test).astype('float32')

    return X_train, y_train, X_val, y_val, X_test, y_test
```

Fonte: Autora (2022).

Embora a função `splitDataset()` permita divisões com diferentes porcentagens, seguindo os passos de Couser et al. (2004), optou-se por utilizar 15% dos dados para

o conjunto de teste, 15% para o conjunto de validação e 70% para o conjunto de treinamento.

A seguir, cada uma das topologias estudadas pode ser criada utilizando-se a função *buildTopology()*, mostrada na Figura 23 e que tem como entrada a estrutura de dados gerada pela função *readInputParams()*, já apresentada neste capítulo. Essa função inicia criando um vetor vazio denominado *topology*, para o qual são adicionadas informações referentes à rede, ou seja, o número de neurônios da camada de entrada, da camada intermediária e da e camada de saída, além da quantidade de camadas intermediárias.

Figura 23 – Função *buildTopology()*

```
def buildTopology(params):
    topology = []

    input_layer = keras.layers.Input(len(params['input_columns']) + 1, name='input')
    topology.append(input_layer)

    hidden_layers = []

    num_neurons = params['num_neurons_per_hidden_layer']

    for i in range(params['num_hidden_layers']):
        inner_layer = keras.layers.Dense(num_neurons, activation=tf.nn.relu, name='inner' + str(i))
        hidden_layers.append(inner_layer)

    topology += hidden_layers

    output_layer = keras.layers.Dense(1, name='output')
    topology.append(output_layer)

    return topology
```

Fonte: Autora (2022).

Finalizada a construção do modelo, por fim é realizado o processo de treinamento da rede, cuja realização é considerada etapa fundamental no desenvolvimento de uma rede neural. Nessa etapa definiu-se a função *trainModel()*, cujas linhas de código são mostradas na Figura 24, e que recebe as características da rede (*params*), o conjunto de treinamento (*train_set*), o conjunto de validação (*val_set*) e o número de ciclos de treinamento (*number_epochs*).

Figura 24 – Função *trainModel()*

```
def trainModel(params, train_set, val_set, number_epochs):
    topology = buildTopology(params)

    model = keras.Sequential(topology)

    model.compile(optimizer='adam', loss='mse', metrics=['mean_absolute_percentage_error'])

    history = model.fit(*train_set, epochs=number_epochs, batch_size=15, validation_data=val_set)

    return model, history
```

Fonte: Autora (2022).

É importante destacar que o parâmetro *number_epochs* indica a quantidade de vezes que o conjunto de validação é utilizado para ajustar tanto os limites de ativação de cada neurônio quanto os pesos de cada ligação da rede neural.

Seguindo o proposto por Couser et al. (2004), o conjunto de treinamento é utilizado em sequência, sempre indo do item 1 até o item n, repetindo o processo na ordem inversa, indo do item n ao item 1.

Para o processo de treinamento em si, foi utilizado o *Keras Sequential* que tem como entrada a topologia criada anteriormente, sendo que os demais parâmetros são configurados por meio da função *model.compile*.

Proposto por Kingma e Ba (2015), o algoritmo Adam Optimizer é uma alternativa ao método iterativo Gradiente Descendente Estocástico para o cálculo e atualização dos pesos da rede. Uma vez que sua documentação já se encontra disponível na biblioteca Keras, o algoritmo Adam é considerado de fácil implementação, além de ser apropriado a problemas que apresentam uma grande quantidade de dados, sendo constantemente associado ao treinamento de redes neurais.

Tanto a avaliação da rede neural durante o treinamento quanto a avaliação final do modelo em relação ao conjunto de teste foram feitas com base no erro quadrático médio. Esse método possibilita que, durante o aprendizado, erros maiores sejam punidos mais intensamente do que os erros menores, induzindo, ajustes mais intensos na rede sempre que os erros são maiores.

Finalizada a execução da função *trainModel()*, a mesma retorna tanto o modelo treinado, quanto o respectivo histórico de treinamento (Figura 24). Quanto ao histórico, o mesmo corresponde ao valor do EQM calculado ao final de cada ciclo de treinamento.

Nota-se que, quanto menor for o valor do parâmetro *batchs*, maior será o custo computacional do processo, pois os pesos e limites serão ajustados um número maior de vezes. Assim, buscando um equilíbrio entre o desempenho e a eficácia do processo de treinamento, arbitrou-se um valor de 15, indicando que a rede será reajustada de 2 a 3 vezes por ciclo de treinamento (*epoch*).

Após a construção, treinamento e validação dos modelos de redes neurais

propostos, definiu-se a função *perGenerationLoss()* (Figura 25), que realiza a geração de relatórios e a respectiva plotagem dos gráficos referentes aos erros quadráticos médio calculados ao longo de cada ciclo de treinamento tanto para o conjunto de treinamento, quanto para o conjunto de validação.

Figura 25 – Função *perGenerationLoss()*

```
def perGenerationLoss(history, params, number_epochs):
    loss_train = history.history['loss']
    loss_val = history.history['val_loss']
    epochs = range(1, number_epochs + 1)

    df = pd.DataFrame(data = {
        'epoch': epochs,
        'loss_train': loss_train,
        'loss_val': loss_val
    })

    df.to_csv(sub_folder_name + '/loss_per_generation.csv', index=False)

    plt.plot(epochs, loss_train, 'g', label='Conjunto de treinamento')
    plt.plot(epochs, loss_val, 'b', label='Conjunto de validação')
    plt.title('Erro quadrático médio por conjunto por geração')
    plt.xlabel('Geração')
    plt.ylabel('Erro quadrático médio')
    plt.legend()
    plt.savefig(sub_folder_name + '/loss_per_generation.png')
    plt.show()
```

Fonte: Autora (2022).

4.2.3 Execução do código

O termo Execução do código refere-se à etapa em que as diferentes funções desenvolvidas são utilizadas em uma estrutura lógica de forma a alcançar o resultado proposto, que no caso, consiste na obtenção das referidas redes neurais devidamente treinadas e cujo desempenho pode ser aferido utilizando-se o conjunto de teste (Figura 26).

Figura 26 – Execução do código

```

param_sets = readInputParams('params.csv')
df_original = pd.read_csv('Banco de Dados - Completo.csv')

number_epochs = 100

num_trials_per_model = 5

parent_folder_name = 'results/' + datetime.datetime.now().strftime("%Y-%m-%d %H-%M-%S")

for params in param_sets:
    dataset = preparaDataset(params, df_original)
    X_train, y_train, X_val, y_val, X_test, y_test = splitDataset(*dataset)

    best_model = None
    best_global_loss = 999999

    for i in range(num_trials_per_model):
        model, history = trainModel(params, (X_train, y_train), (X_val, y_val), number_epochs)

        global_loss = model.evaluate(*dataset, return_dict=True)['loss']

        if global_loss < best_global_loss:
            best_global_loss = global_loss
            best_model = model

    sub_folder_name = parent_folder_name + '/' + params['name']

    best_model.save(sub_folder_name + '/model.keras')
    #visualizer(best_model, filename=sub_folder_name + '/model', format='png', view=False)

    generateReports(best_model, history, dataset, (X_train, y_train, X_val, y_val, X_test, y_test), params, number_epochs)

```

Fonte: Autora (2022).

O número de ciclos de treinamento aplicado corresponde a 100, cujo valor foi arbitrado de forma a se evitar o *Overfitting*, situação na qual a rede está tão treinada que passa a memorizar os dados, perdendo sua capacidade fundamental de generalização. Segundo Brownlee (2020), um número elevado de ciclos de treinamento propicia o surgimento de *Overfitting*, ao passo que um número baixo pode gerar *Underfitting*, ou seja, a rede não adquire a capacidade de generalização.

Uma vez que o processo de construção de uma rede neural possui elementos estocásticos, seguindo os passos de Brownlee (2020), visando reduzir a possibilidade da sequência de treinamento levar o ajuste da rede neural a estagnar em mínimos locais, adicionou-se a variável `num_trials_per_model` que força o processo de treinamento a ser repetido, buscando identificar a rede cujo treinamento resultou no melhor desempenho na avaliação do conjunto de teste. Neste caso, o valor utilizado para essa variável foi de 5 tentativas para cada uma das topologias estudadas.

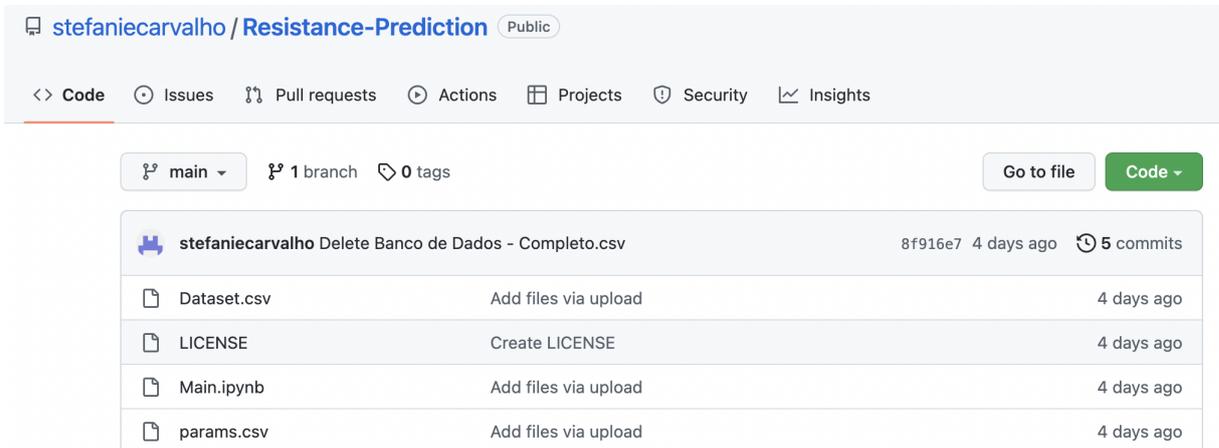
4.2.4 Repositório *Resistance Prediction*

Com o intuito de disponibilizar o programa para trabalhos futuros, utilizou-se a plataforma de desenvolvimento *GitHub* para realizar o armazenamento nas nuvens do código fonte desenvolvido.

O repositório criado, *Resistance Prediction* (Figura 27), contém o código

principal (Main.ipynb), os dados referentes às topologias dos modelos propostos (params.csv), o conjunto de dados utilizado (Dataset.csv) e a licença inserida, a qual é descrita no arquivo LICENSE.

Figura 27 – Repositório *Resistance Prediction*



Fonte: Carvalho (2022).

Optou-se pelo uso da plataforma *GitHub* pelo fato da mesma já ser amplamente utilizada por desenvolvedores de diversas áreas como a de *Machine Learning*, além de contar com uma série de atributos que contribuem para a alteração e o aperfeiçoamento do código gerado, tais como:

- Possibilidade de baixar uma cópia do arquivo completo, bem como a adição de eventuais modificações realizadas;
- Possibilidade de adicionar a versão modificada ao repositório original com a inserção de comentários explicativos.

Dessa forma, a rede neural desenvolvida tem ainda a possibilidade de ser continuamente alterada e aperfeiçoada. Além disso, o conjunto de dados utilizado no desenvolvimento pode ser ampliado a medida em que informações sobre novas embarcações planantes puderem ser encontradas e adicionadas ao arquivo.

5 RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados e discutidos os resultados provenientes do estudo de cada um dos doze modelos de redes neurais estudados neste trabalho. Os resultados foram divididos em quatro categorias, a primeira trata da avaliação das curvas de aprendizado dos modelos, a segunda traz uma análise geral do desempenho dos modelos, enquanto a terceira aborda a análise paramétrica para as topologias de redes neurais utilizadas. Por último, a quarta categoria apresenta um estudo de caso realizado com três embarcações extraídas do conjunto de dados.

É importante destacar que, conforme discutido no Capítulo 4, devido ao carácter estocástico do processo de construção de uma rede neural, em todos os casos, o processo de treinamento foi sempre repetido 5 vezes. Assim, todos os resultados apresentados neste capítulo referem-se sempre ao processo de treinamento que resultou no menor EQM aferido na análise do conjunto de teste.

5.1 CURVAS DE APRENDIZADO

As curvas de aprendizado trazem informações sobre a performance e a capacidade de generalização da rede neural ao longo de cada ciclo do treinamento.

Seguindo o exposto por Brownlee (2019), o aprendizado de uma rede neural pode ser mostrado plotando-se em um único gráfico, a curva de erro (*loss*) por ciclo de treinamento (*epochs*) tanto para o conjunto de treinamento, quanto para o conjunto de validação.

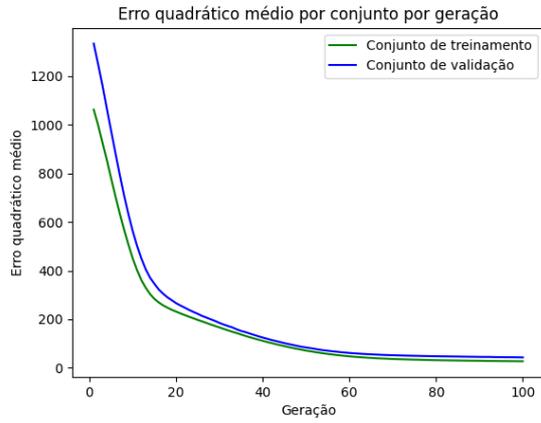
Enquanto a curva para o conjunto de treinamento representa o quão bem a rede está aprendendo, a curva para o conjunto de validação retrata o quão bem a rede está sendo capaz de generalizar o problema em questão. Nesse contexto, a performance dos doze modelos treinados pode ser vista nas Figuras 28 e 29.

A análise das curvas de aprendizado referentes a cada um dos modelos estudados considerou os seguintes aspectos visuais:

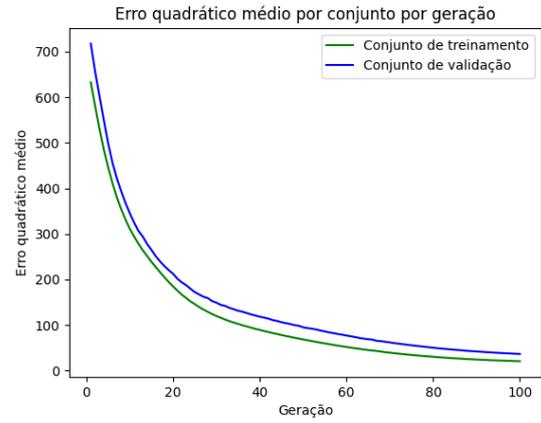
1. A curva de treinamento decresce até um certo ponto e depois se estabiliza;
2. A curva de validação decresce até um certo ponto e depois se estabiliza a uma certa distância da curva de treinamento, gerando o chamado espaço de generalização;
3. A curva de treinamento está abaixo da curva de validação.

Esses aspectos possibilitam compreender se o comportamentos dos modelos treinados está super ajustado (*overfitted*), sub ajustado (*underfitted*) ou simplesmente bem ajustado (*good fitted*).

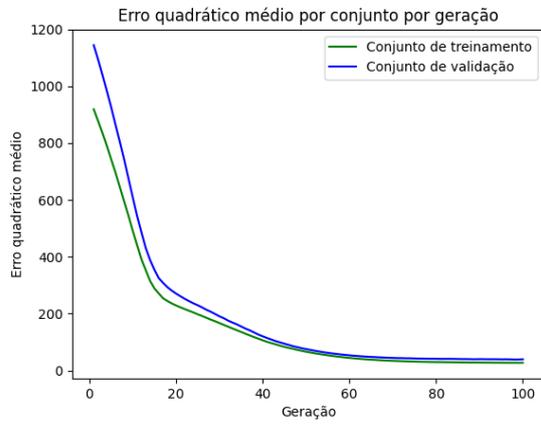
Figura 28 – Curvas de aprendizado - M1 a M6



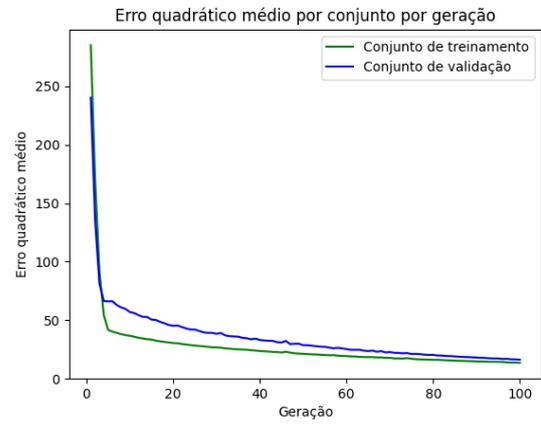
(a) M1



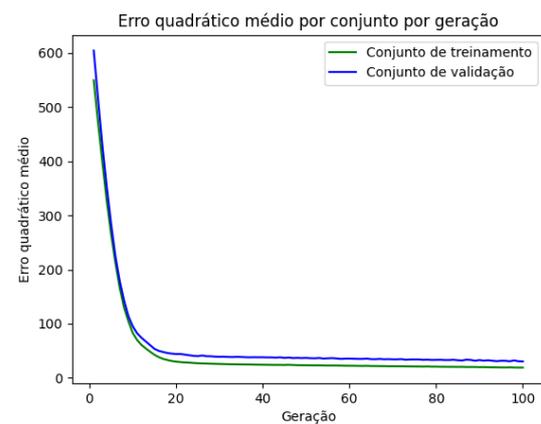
(b) M2



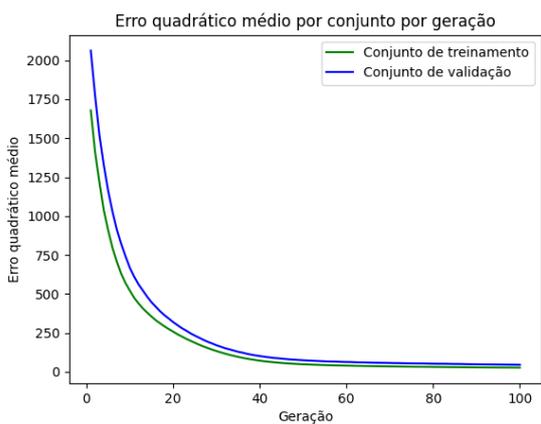
(c) M3



(d) M4



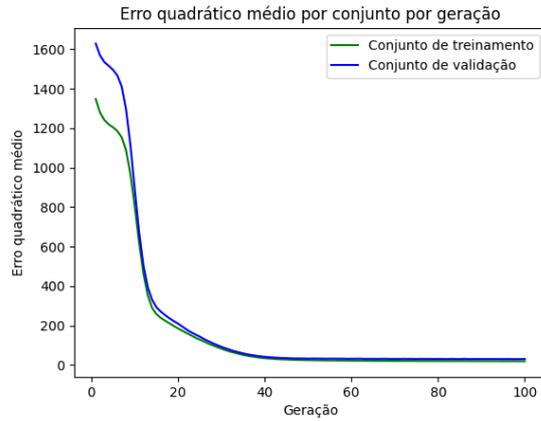
(e) M5



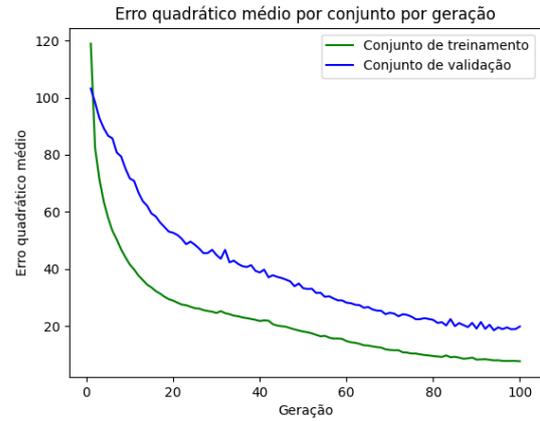
(f) M6

Fonte: Autora (2022).

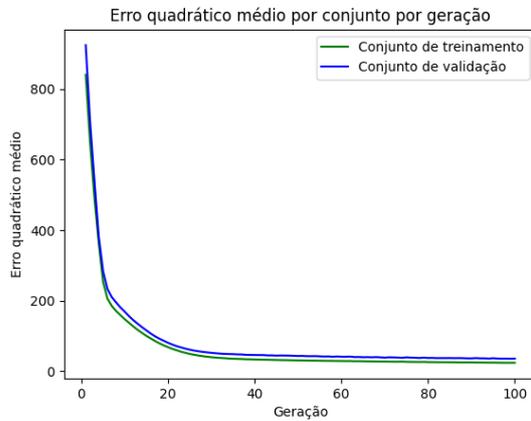
Figura 29 – Curvas de aprendizado - M7 a M12



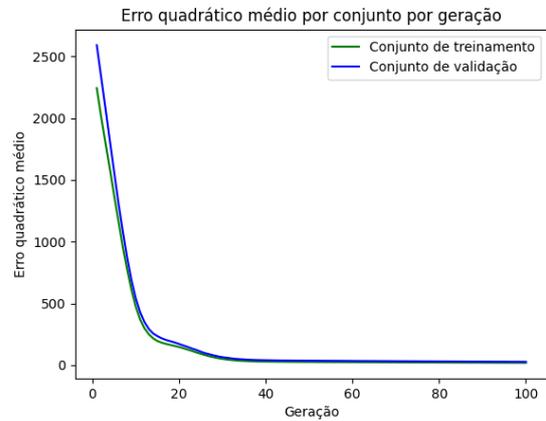
(a) M7



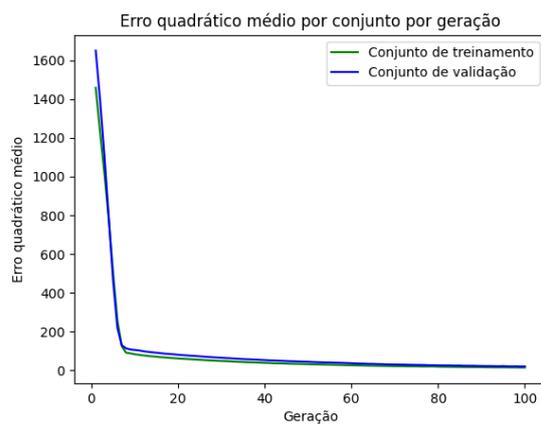
(b) M8



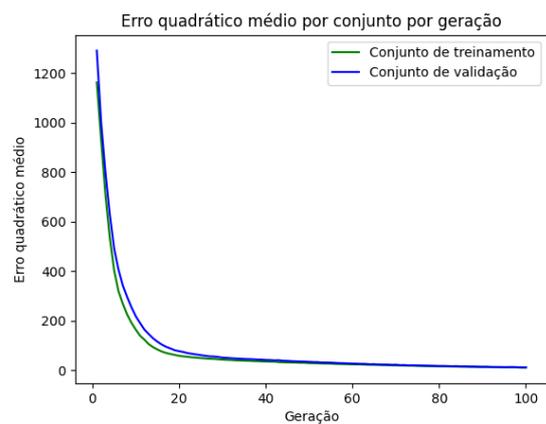
(c) M9



(d) M10



(e) M11



(f) M12

Fonte: Autora (2022).

Com exceção do modelo M8, os demais modelos apresentaram características de modelos bem ajustados (*good fitted*) com curvas de treinamento e validação que decrescem durante uma quantidade de ciclos de treinamento e depois se estabilizam.

O número de ciclos a partir do qual o treinamento estabiliza varia de 20 (modelos M12, M11, M9 e M5) a 60 (modelos M4, M3 e M1), chegando a 100 para o modelo M2, indicando que a maioria dos modelos apresenta pouca evolução após 60 ciclos de treinamento.

Nota-se que o modelo M8 é único que apresentou uma parada antecipada (*Early Stopping*), uma vez que a curva de treinamento continua a decrescer ao longo dos ciclos, enquanto a curva de validação se distancia da curva de treinamento, indicando a necessidade de um número maior de ciclos de treinamento. O motivo pelo qual o modelo M8 apresentou esse comportamento não pôde ser compreendido.

Em geral, os resultados mostraram que os modelos mais complexos, com mais conexões neurais, necessitaram de um número menor de ciclos de treinamento para atingirem a estabilização (Tabela 5). Isso pode ocorrer pois a pouca quantidade de conexões neurais existentes em modelos mais simples exige parâmetros mais precisos para generalizar corretamente o problema, ou ainda, podem ser insuficientes para identificar todas as tendências existentes no conjunto de dados.

Tabela 5 – Número de conexões e de ciclos de treinamento até à estabilização do treinamento de cada modelo

Modelo	Número de conexões	Número de ciclos até à estabilização
M1	21	60
M2	35	100
M3	30	60
M4	60	60
M5	50	20
M6	70	40
M7	75	40
M8	119	>100
M9	98	20
M10	126	40
M11	147	20
M12	207	20

Fonte: Autora (2022).

Por fim, para todos os modelos avaliados é possível verificar que a curva de validação está acima da curva de treinamento. Esse resultado é coerente com o fato de que o uso de uma rede neural para estimar resultados de casos não utilizados no treinamento tende a resultar em um erro maior do que quando são avaliados os dados utilizados no treinamento. Nesse sentido, a diferença entre o último EQM de validação e o último EQM de treinamento corresponde ao termo conhecido como espaço de generalização que, quando positivo, corrobora para o comportamento *good fitted* da rede neural.

Na Tabela 6 são apresentados os erros quadráticos médios no último ciclo de treinamento, tanto para o conjunto de treinamento quanto para o conjunto de validação, bem como o espaço de generalização (EG) resultante.

Tabela 6 – EQM para o último ponto das curvas de treinamento e validação e valor referente ao espaço de generalização (EG)

Modelo	EQM treinamento no 100° ciclo (a)	EQM validação no 100° ciclo (b)	EG (b - a)
M1	27,25	43,19	15,94
M2	20,53	36,53	15,99
M3	27,40	39,61	12,21
M4	13,49	16,17	2,68
M5	19,30	30,62	11,33
M6	27,57	45,62	18,05
M7	19,27	30,81	11,54
M8	7,62	19,81	12,19
M9	24,16	36,09	11,92
M10	19,52	27,65	8,12
M11	14,47	20,96	6,48
M12	11,16	12,23	1,07

Fonte: Autora (2022).

5.2 ANÁLISE GERAL DOS MODELOS

Após avaliar o comportamento dos modelos de redes neurais ao longo do processo de treinamento (curva de aprendizado), avaliou-se o erro quadrático médio, calculado para o conjunto de teste de cada modelo, visando verificar o desempenho da rede na estimativa de resistência ao avanço de casos nunca utilizados no procedimento de treinamento.

Para facilitar a visualização e a análise, os resultados referentes à aplicação dos diferentes modelos de redes neurais desenvolvidos neste trabalho e aplicados ao conjunto de teste, foram reunidos na Figura 30.

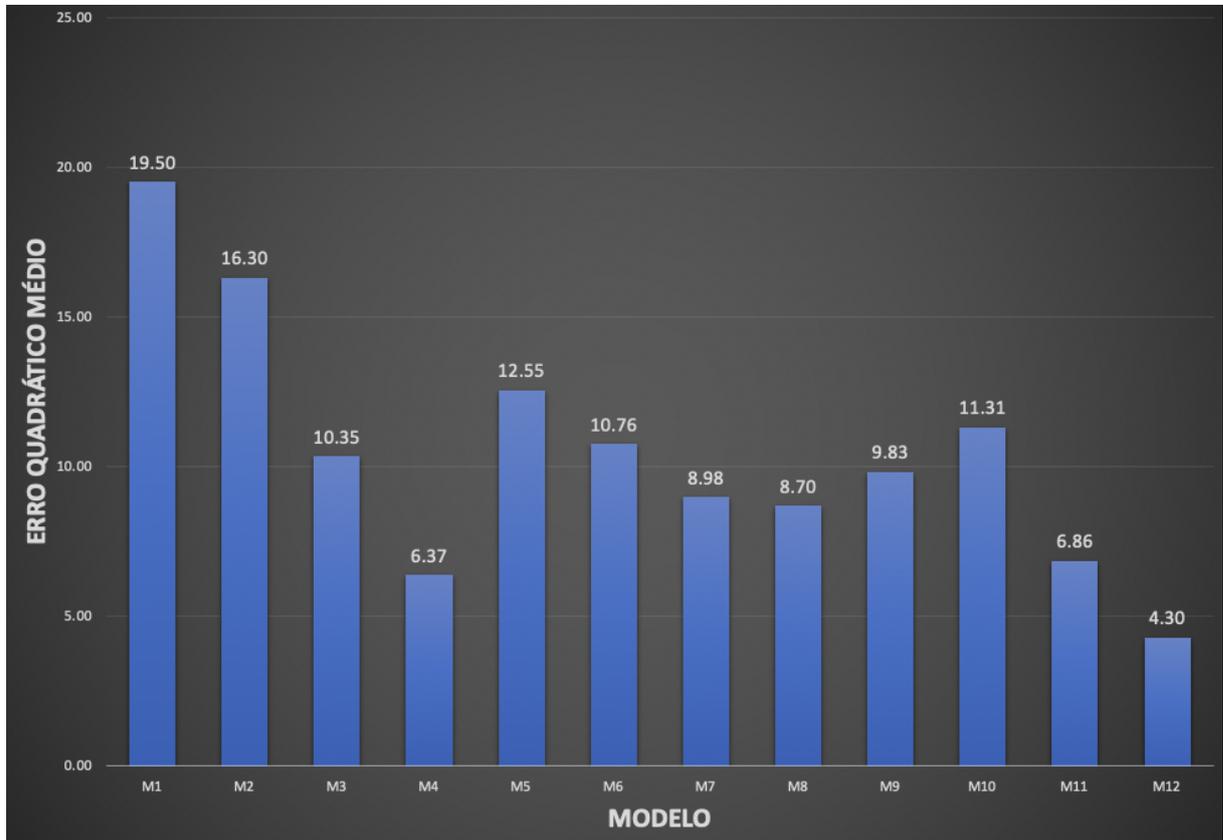
Analisando os resultados obtidos, é possível observar que os modelos que apresentaram o menor EQM foram os modelos M12, M4 e M11, enquanto o modelo com maior EQM foi o modelo M1.

Dessa forma é possível inferir que o aumento da complexidade dos modelos, inserindo mais camadas e mais neurônios contribui para a redução do EQM. No entanto, casos como o modelo M4 chamam a atenção, pois apresentam baixo EQM aliado a uma topologia mais simples e eficiente.

Como será visto adiante, o número de camadas internas apresenta grande influência sobre o resultado final da rede neural. O modelo M1 é aquele que dentre

todos possui o menor número de parâmetros de entrada, de camadas internas e de neurônios, o que contribui para um modelo mais improvável de generalizar o comportamento da resistência ao avanço.

Figura 30 – Erro quadrático médio do conjunto de teste por modelo



Fonte: Autora (2022).

Para evidenciar essa conclusão, observa-se que os modelos M1 e M12, além de apresentarem, respectivamente, o menor e o maior grau de complexidade, apresentaram, respectivamente, o maior e o menor EQM aferido para o conjunto de teste.

No entanto, a Figura 30 evidencia que o EQM não segue estritamente a variação da complexidade do modelo, o que pode indicar uma possível influência dos parâmetros da embarcação que foram selecionados como parâmetros de entrada em cada uma das topologias estudadas.

5.3 ANÁLISE PARAMÉTRICA

Nesta seção é apresentada a análise paramétrica realizada a partir do desempenho dos doze modelos de redes neurais estudados. O objetivo desta análise é determinar, se possível, a influência de cada um desses parâmetros no desempenho da rede neural, o qual é aferido após o processo de treinamento.

Os parâmetros analisados referem-se às características da rede referente a cada um dos modelos estudados, ou seja, o número de parâmetros de projeto utilizados como entrada (grupo), o número de camadas internas e o número de neurônios das camadas internas.

Na Tabela 7 são apresentados os resultados da análise paramétrica realizada, onde foram atribuídos os valores 1 ou 0 para indicar a presença ou não do parâmetro esperado no modelo avaliado. Essa análise teve como premissa a hipótese de que modelos com EQM menores seriam mais complexos.

Assim, acredita-se que os seis primeiros modelos a apresentarem os menores EQM seriam aqueles com duas camadas internas. Os parâmetros de entrada e o número de neurônios nas camadas internas também contribuem para a hierarquização dos modelos esperados em ordem crescente de EQM. Em outras palavras, espera-se que quanto mais complexo o modelo, menor será o seu EQM. Essa ideia é confrontada com o ranking obtido durante o processo de verificação realizado após o treinamento ser concluído, o qual considerou a aplicação da rede neural ao conjunto de testes não utilizado durante o treinamento.

Tabela 7 – Análise paramétrica dos modelos

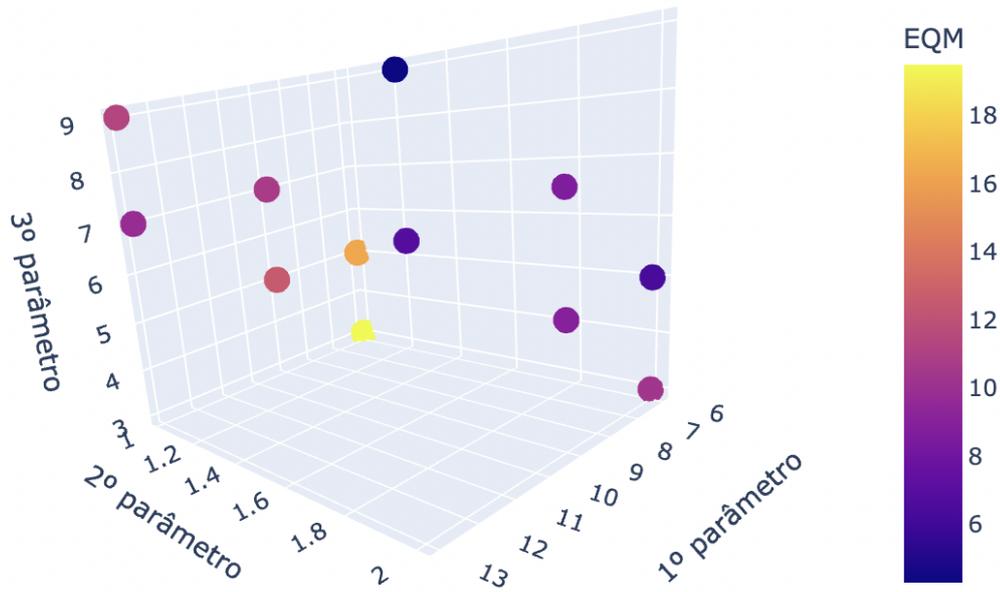
Ranking		Primeiro parâmetro			Segundo parâmetro			Terceiro parâmetro		
Modelo	EQM	Grupo	Grupo esperado	Relação	No de camadas internas	No de camadas internas esperado	Relação	No de neurônios na camada interna	No de neurônios na camada interna esperado	Relação
M12	4,30	G3	G3	1	2	2	1	9	9	1
M4	6,37	G1	G3	0	2	2	1	5	7	0
M11	6,86	G3	G2	0	2	2	1	7	7	1
M8	8,70	G2	G2	1	2	2	1	7	5	0
M7	8,98	G2	G1	0	2	2	1	5	5	1
M9	9,83	G3	G1	0	1	2	0	7	5	0
M3	10,35	G1	G3	0	2	1	0	3	9	0
M6	10,76	G2	G3	0	1	1	1	7	7	1
M10	11,31	G3	G2	0	1	1	1	9	7	0
M5	12,55	G2	G2	1	1	1	1	5	5	1
M2	16,30	G1	G1	1	1	1	1	5	5	1
M1	19,50	G1	G1	1	1	1	1	3	3	1
			Soma	5		Soma	10		Soma	7
			Influência [%]	41,67		Influência [%]	83,33		Influência [%]	58,33

Fonte: Autora (2022).

A influência final de cada parâmetro é dada pelo percentual resultante de relações equivalentes a 1. Os resultados indicam que o número de camadas internas tem maior influência no desempenho da rede (EQM), alcançando 83,33% de influência. Já os parâmetros de entrada (grupo) e número de neurônios na camada interna apresentam uma influência menor sobre o desempenho da rede, com 41,67% e 58,33%, respectivamente.

Para ilustrar essa significativa influência do número de camadas internas sobre o desempenho final da rede neural, nota-se que o EQM de todos os modelos com duas camadas internas é inferior ao EQM de modelos que apresentam somente uma camada interna (Figura 31), corroborando a análise paramétrica realizada.

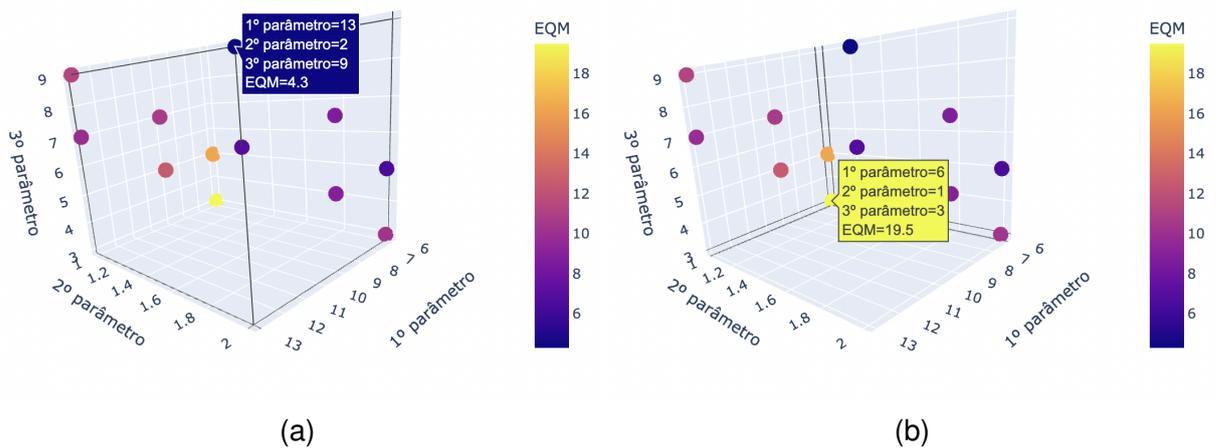
Figura 31 – Distribuição espacial dos EQM dos modelos neurais estudados



Fonte: Autora (2022).

Já na Figura 32 são destacados os posicionamentos dos modelos limites M1 e M12 que representam o modelo mais simples (6 entradas, 1 camada interna e 3 neurônios nas camadas internas) e o mais complexo (13 entradas, 2 camadas internas e 9 neurônios nas camadas internas), respectivamente.

Figura 32 – Posição espacial: modelo M12 (a) e modelo M1 (b)



Fonte: Autora (2022).

A partir da observação das curvas de aprendizado e da análise paramétrica dos doze modelos estudados, é possível concluir que os dois modelos que apresentaram menor EQM foram os modelos M12 e M4, enquanto o modelo que apresentou o maior EQM foi o modelo M1.

5.4 CURVAS DE RESISTÊNCIA POR VELOCIDADE

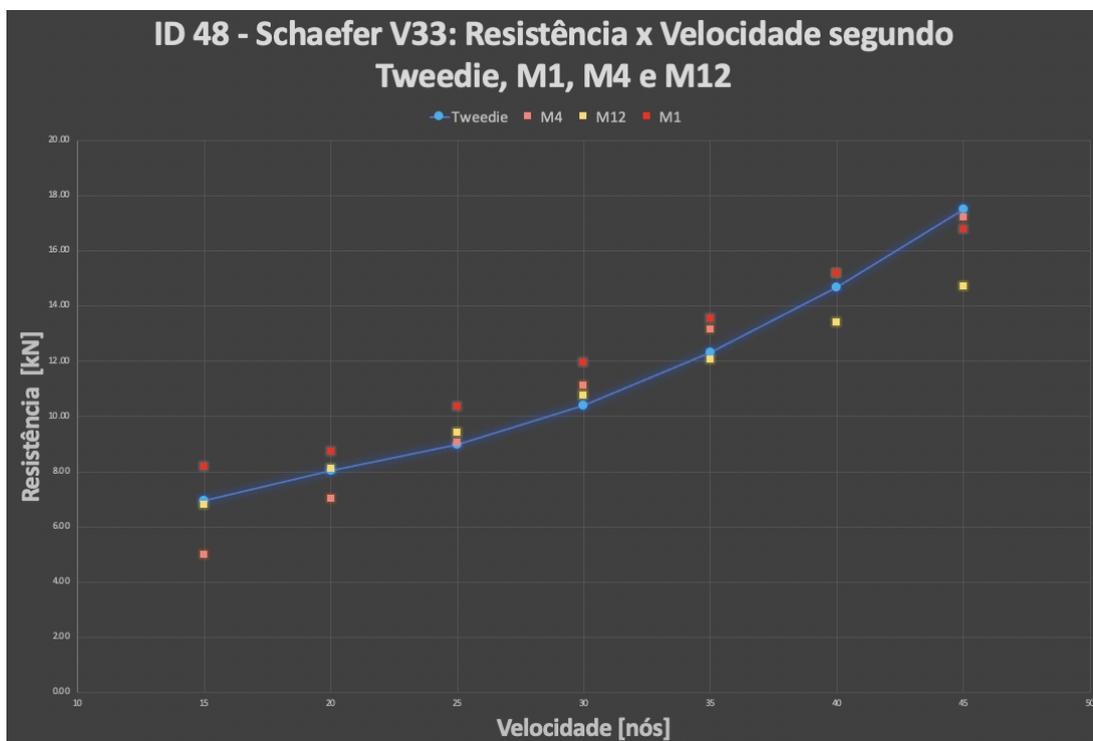
A fim de comparar e visualizar os resultados referentes à estimativa da resistência ao avanço considerando tanto os principais modelos neurais desenvolvidos neste trabalho (M12, M4 e M1) quanto a implementação tradicional do método de Savitsky (1964) realizada por Tweedie (2011), foram analisadas três embarcações do conjunto de dados mostrado no Apêndice A.

A seleção das embarcações buscou contemplar uma embarcação no limite superior de deslocamento (ID 31 - Azimut 75), uma embarcação com deslocamento intermediário (ID 5 - BG54) e, por fim, uma embarcação no limite inferior de deslocamento (ID 48 - Schaefer V33).

Embora a análise realizada foque nas três embarcações selecionadas, o conjunto completo de resultados referente ao cálculo das resistências ao avanço para todas as 54 embarcações utilizando os modelos M4, M12 e M1 pode ser visto nos Apêndices H e M deste trabalho.

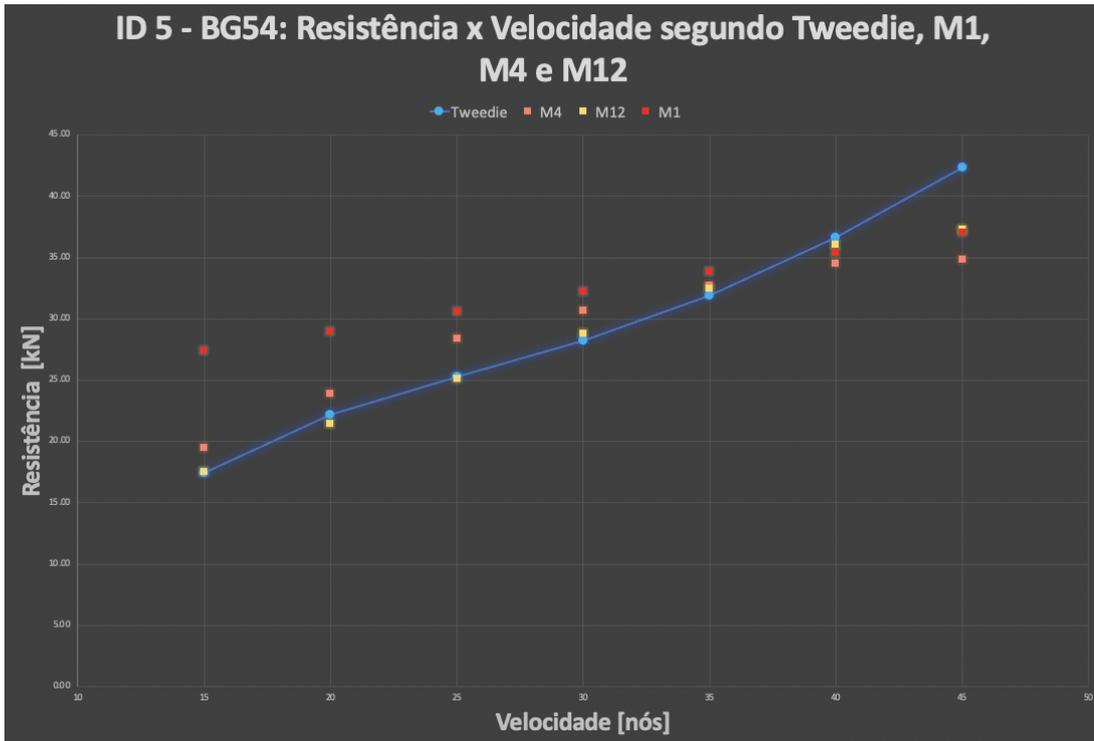
A seguir são mostradas a análise da embarcação Schaefer V33 (Figura 33), embarcação BG54 (Figura 34) e embarcação Azimut 75 (Figura 35).

Figura 33 – Curva de resistência por velocidade para Schaefer V33



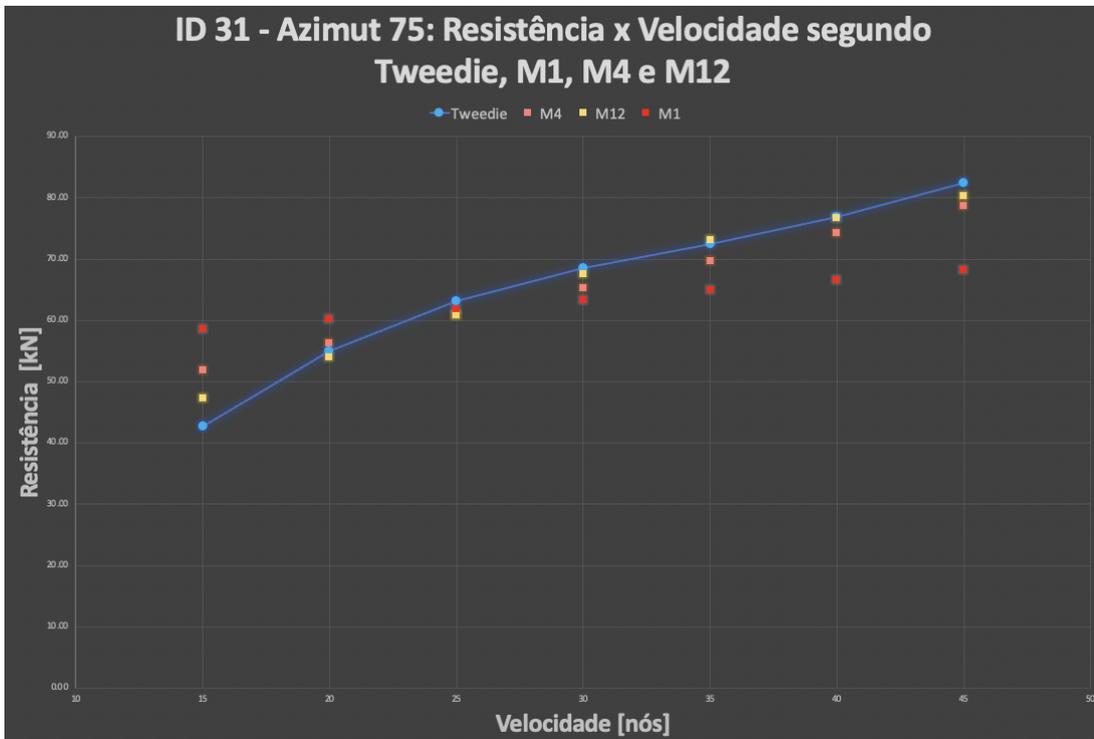
Fonte: Autora (2022).

Figura 34 – Curva de resistência por velocidade para BG54



Fonte: Autora (2022).

Figura 35 – Curva de resistência por velocidade para Azimut 75



Fonte: Autora (2022).

De maneira geral, todas as curvas plotadas, indiferente da embarcação avaliada, apresentam um comportamento semelhante. Os resultados obtidos usando os modelos M12 e M4 estão mais próximos dos resultados esperados do que aqueles obtidos utilizando o modelo M1.

Além disso, os resultados obtidos nessa etapa permitem verificar, a um só tempo, duas afirmações feitas por Tancredi (2008). A primeira consiste no fato de que uma rede neural inicialmente captura o valor médio do conjunto de dados, capturando mais tendências a medida que a complexidade da rede aumenta e/ou uma maior quantidade de ciclos de treinamento é realizada.

A segunda, consiste no fato de que uma rede neural tem mais facilidade para capturar os efeitos na região central dos dados, tendo maior dificuldade para capturar as tendências associadas aos limites dos dados existentes no conjunto de treinamento.

6 CONCLUSÕES

O presente trabalho objetivou utilizar algoritmos de *Machine Learning*, mais especificamente redes neurais artificiais, para solucionar um problema típico da Engenharia Naval, estimar a resistência ao avanço de embarcações. Conforme visto na Introdução, observou-se uma lacuna no que se refere ao uso de redes neurais para estimar a resistência ao avanço de embarcações planantes. Nesse contexto, este trabalho apresentou o desenvolvimento de uma rede neural implementada em Python destinada a estimar a resistência de cascos planantes, cujo conjunto de treinamento foi gerado utilizando-se a implementação proposta por Tweedie (2011) do método de Savitsky (1964).

A fim de cumprir o objetivo geral proposto, foram planejados doze diferentes modelos de redes neurais, para os quais variou-se tanto os parâmetros referentes à topologia da rede (número de camadas internas e número de neurônios das camadas internas) quanto o número de entradas da rede (parâmetros relevantes para o projeto de embarcações planantes).

Essas diferentes topologias foram avaliadas em relação à sua capacidade de generalização e a qualidade das previsões que realizam, sempre utilizando como medida de desempenho o erro quadrático médio calculado para um conjunto de teste não utilizado no treinamento.

O conjunto de dados utilizados neste trabalho, contou com informações referentes a 54 embarcações planantes reais, cujos dados foram obtidos da literatura disponível. Embora em alguns casos, por falta de informação, tenha sido necessário realizar a estimativa de alguns parâmetros, acredita-se que essas estimativas não interferiram nos resultados das redes neurais, uma vez que os resultados esperados, calculados por meio da planilha de Tweedie (2011), também consideraram os mesmos parâmetros estimados.

O código proposto foi desenvolvido em Python e utiliza bibliotecas sabidamente robustas e eficientes para o tratamento dos dados, a definição da topologia das redes, o treinamento, a validação e a plotagem de resultados.

Todos os modelos estudados passaram por 100 ciclos de treinamento, utilizando lotes de 15 pontos pertencentes sempre ao mesmo conjunto de treinamento. Além disso foram utilizadas funções de ativação do tipo ReLU para as camadas internas e do tipo linear para a camada de saída. Sendo que a proporção utilizada para a divisão do conjunto de dados consistiu em 70% para o conjunto de treinamento, 15% para o de validação e 15% para o de teste.

Os resultados mostraram que, de maneira geral, a partir de 60 ciclos de

treinamento, as redes estudadas foram capazes de identificar padrões, permitindo estimar a resistência ao avanço de um casco planante com diferentes graus de precisão, com a inexplicável exceção da rede M8, cujo comportamento não pôde ser compreendido.

Considerando a análise utilizando exclusivamente o conjunto de teste, embora o modelo M12 de tamanho 13x9x1 tenha apresentado o menor erro quadrático médio (4,30), considera-se que o modelo M4 de tamanho 6x5x1, obteve um desempenho melhor, pois, além de apresentar o segundo menor erro quadrático médio (6,37), conseguiu estimar a resistência ao avanço utilizando uma quantidade menor de parâmetros de entrada do que o modelo M12. Por outro lado, o modelo M1, justamente por ser o mais simples (com um menor número de ligações neurais), foi aquele que apresentou os maiores valores de erro quadrático médio.

Por fim, os resultados mostraram que o parâmetro que mais influenciou no desempenho das redes estudadas foi o número de camadas internas do modelo, sendo que as redes com duas camadas internas apresentaram sempre um erro quadrático médio inferior às redes com apenas uma camada interna.

Em virtude dos aspectos relatados, entende-se que este trabalho alcançou os objetivos propostos dentro dos limites estabelecidos para a pesquisa. No entanto, para trabalhos futuros sugere-se: a ampliação do conjunto de dados; uma avaliação paramétrica mais profunda quanto aos parâmetros referentes ao projeto de uma embarcação planante, incluindo, por exemplo, a presença de apêndices; além da avaliação de outros algoritmos de treinamento e de outros tipos de funções de ativação.

REFERÊNCIAS

ABRAMOWSKI, T. Application of artificial intelligence methods to preliminary design of ships and ship performance optimization. **Naval Engineers Journal**, v. 125, n. 3, p. 101–112, 2013.

ALLEBRANDT JUNIOR, A. **Desenvolvimento de ferramenta computacional para a análise da resistência ao avanço e da estabilidade de cascos planantes**. 2021. Trabalho de Conclusão de Curso (Graduação em Engenharia Naval) — Centro Tecnológico de Joinville, Universidade Federal de Santa Catarina, Joinville, 2021.

BERTRAM, V. **Practical ship hydrodynamics**. Oxford, England: Butterworth-Heinemann, 2000.

BROWNLEE, J. **How to use learning curves to diagnose machine learning model performance**. 2019. Disponível em: <https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/>. Acesso em: 17 nov. 2022.

BROWNLEE, J. **Basic concepts in machine learning**. 2020. Disponível em: <https://machinelearningmastery.com/basic-concepts-in-machine-learning/>. Acesso em: 17 nov. 2022.

BRUNER, E. M. **Desenvolvimento de ferramenta computacional educacional para projeto estrutural de embarcações deslocantes com base na norma ABS**. 2019. Trabalho de Conclusão de Curso (Graduação em Engenharia Naval) — Centro Tecnológico de Joinville, Universidade Federal de Santa Catarina, Joinville, 2019.

CARVALHO, R. S. **Resistance prediction**. 2022. Disponível em: <https://github.com/stefaniecarvalho/Resistance-Prediction>. Acesso em: 20 nov. 2022.

CEPOWSKI, T. The prediction of ship added resistance at the preliminary design stage by the use of an artificial neural network. **Ocean Engineering**, v. 195, p. 106657, 2020.

COUSER, P. et al. Artificial neural networks for hull resistance prediction. *In: Proceedings of the CONFERENCE COMPIT*. p. 391–402, 2004. Disponível em: https://www.academia.edu/2384082/Artificial_neural_networks_for_hull_resistance_prediction. Acesso em: 27 out. 2022.

DIAS NETO, L. A. V.; ONETY, R. E.; TANCREDI, T. P. Conforto ou velocidade? Uma visão multiobjetivo do clássico dilema para o projeto de lanchas. *In: Anais do congresso brasileiro interdisciplinar em ciência e tecnologia. Anais [...] Diamantina, MG: UFVJM*. 2021. Disponível em: <https://www.even3.com.br/anais/cobicet/390227-conforto-ou-velocidade-uma-visao-multiobjetivo-do-classico-dilema-para-o-projeto-de-lanchas/>. Acesso em: 27 ago. 2022.

EVANS, J. H. Basic design concepts. **Journal of the American Society for Naval Engineers**, v. 71, n. 4, p. 671–678, 1959.

FALTINSEN, O. M. **Hydrodynamics of high-speed marine vehicles**. United States: Cambridge, 2005.

FAUSETT, L. **Fundamentals of neural networks: architectures, algorithms and applications**. United States: Prentice Hall, 1993.

HAYKIN, S. **Neural networks and learning machines**. 3. ed. Nova Jersey, United States: Pearson Education, 2008.

HEATON, J. **The number of hidden layers**. 2017. Disponível em: <https://www.heatonresearch.com/2017/06/01/hidden-layers.html>. Acesso em: 20 nov. 2022.

HOLTROP, J.; MENNEN, G. G. J. An approximate power prediction method. **International Shipbuilding Progress**, v. 29, n. 335, p. 166–170, 1982.

JUAN, N. P.; VALDECANTOS, V. N. Review of the application of artificial neural networks in ocean engineering. **Ocean Engineering**, v. 259, p. 111947, 2022.

KAVLAKOGLU, E. **AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the difference?** 2020. Disponível em: <https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks>. Acesso em: 12 nov. 2022.

KINGMA, D. P.; BA, J. L. Adam: a method for stochastic optimization. *In: Proceedings of the THIRD INTERNATIONAL CONFERENCE ON LEARNING REPRESENTATIONS*. p. 1–13, 2015.

KRIESEL, D. **A brief introduction to neural networks**. 2007. Disponível em: <http://www.dkriesel.com>. Acesso em: 20 ago. 2022.

KUKNER, A.; YASA, A. M. High speed planning hulls resistance prediction methods and comparasion. *In: Proceedings of the FIRST INTERNATIONAL SYMPOSIUM ON NAVAL ARCHITECTURER AND MARITIME (INT-NAM)* . p. 201–208, 2011.

LAMB, T. **Ship design and construction**. Michigan, United States: The Society of Naval Architects and Marine Engineers, 2003.

LEME, F. **Análise da influência do parâmetro corda/diâmetro na eficiência de hélices utilizando teoria de linha de sustentação**. 2021. Trabalho de Conclusão de Curso (Graduação em Engenharia Naval) — Centro Tecnológico de Joinville, Universidade Federal de Santa Catarina, Joinville, 2021.

MARIN, G. **Estudo Paramétrico de resistência ao avanço de uma embarcação de planeio: análise método de Savitsky**. 2015. Trabalho de Conclusão de Curso (Graduação em Engenharia Naval) — Centro Tecnológico de Joinville, Universidade Federal de Santa Catarina, Joinville, 2015.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biology**, v. 5, n. 1/2, p. 115–133, 1943.

MCKINSEY. **The state of AI in 2021**. 2021. Disponível em: <https://www.mckinsey.com/capabilities/quantumblack/our-insights/global-survey-the-state-of-ai-in-2021>. Acesso em: 20 nov. 2022.

MOLLAND, A. **The maritime engineering reference book**. Burlington, United States: Butterworth-Heinemann, 2008.

NÉIA, E. C. **Modelo paramétrico para resistência ao avanço, comportamento em mar e estabilidade de navios petroleiros baseado em regressão matemática e redes neurais artificiais**. Trabalho de Conclusão de Curso (Graduação em Engenharia Naval) — Centro Tecnológico de Joinville, Universidade Federal de Santa Catarina, Joinville, 2016.

PAPANIKOLAOU, A. **Ship design: methodologies of preliminary design**. Atenas: Springer, 2014.

PRABOWO, A. R. et al. Effect of hull design variations on the resistance profile and wave pattern: A case study of the patrol boat vessel. **Journal of Engineering Science and Technology**, v. 17, n. 1, p. 106–126, 2022.

PRAMODITHA, R. **How to choose the right activation function for neural networks**. 2019. Disponível em: <https://towardsdatascience.com/how-to-choose-the-right-activation-function-for-neural-networks-3941ff0e6f9c>. Acesso em: 10 nov. 2022.

RIBEIRO, H. J. C. **Equilíbrio dinâmico de cascos planadores**. 2002. Dissertação (Mestrado em Engenharia Naval) — COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2002.

SAVITSKY, D. Hydrodynamic design of planing hulls. **Marine Technology**, v. 1, n. 1, p. 71–95, 1964.

SPAHIC, B. **Python ohne vorkenntnisse: Innerhalb von 7 tagen ein neuronales netz programmieren**. Deutschland: PBT, 2020.

TANCREDI, T. P. **Otimização multidisciplinar distribuída aplicada a projetos de engenharia**. 2008. Tese (Doutorado em Engenharia Naval e Oceânica) — Escola Politécnica, Universidade de São Paulo, São Paulo, 2008.

TRINDADE, J. **Hidrodinâmica e propulsão: Engenharia de Máquinas Marítimas**. Portugal: ENIDH, 2012.

TWEEDIE, D. **Savitsky power prediction spreadsheet**. 2011. Disponível em: <https://www.boatdesign.net/threads/savitsky-power-prediction.2187/page-9#post-449328>. Acesso em: 15 jun. 2022.

VANDERPLAS, J. **Python data science handbook**. Sebastopol, United States: O'Reilly, 2017.

WATSON, D. G. M. **Practical ship design**. Oxford, England: Elsevier Science, 1998.

YANG, Y. et al. Research on accurate prediction of the container ship resistance by rbfnn and other machine learning algorithms. **Journal of Marine Science and Engineering**, v. 9, n. 4, p. 376, 2021.

APÊNDICE A - IDENTIFICAÇÃO DAS EMBARCAÇÕES ANALISADAS

Tabela 8 – Identificação das embarcações do conjunto de dados

ID	Nome	ID	Nome
1	BG60	28	S8
2	BGX60	29	Azimut 70
3	BGX70	30	70 Seajet
4	BG42	31	Azimut 75
5	BG54	32	Azimut 80
6	BG62	33	Fly 58
7	BG27	34	Ferretti 53
8	Aquarama	35	Alfamarine 60
9	Aquariva Super	36	Portofino 750 Spider
10	ISEO	37	Sabreline 36
11	Rivamare	38	Sun Sport 240
12	Dolceriva	39	Sun Sport 220
13	56´Rivale	40	Sun Sport 230
14	68´Diable New	41	Sea Ray 240 Sundeck
15	88´Florida	42	Sundancer 265
16	54 Aquarius	43	Sundancer 320
17	66´ Ribelle	44	Sundancer 320 OB
18	21 Dolce Vita	45	Sundancer 370 OB
19	76´Bahamas	46	Oyster 34
20	Verve 47	47	Oyster 36
21	Verve 42	48	Schaefer V33
22	Azimut 42	49	Schaefer 303
23	Atlantis 35	50	Schaefer 400
24	Azimut 39	51	Schaefer 365
25	Azimut 47	52	Schaefer 375 HT
26	Atlantis 55	53	Schaefer 510 S
27	Fly 38	54	Schaefer 600

Fonte: Autora (2022).

APÊNDICE B - CARACTERÍSTICAS DAS EMBARCAÇÕES ANALISADAS

Tabela 9 – Conjunto de dados (Parte I)

ID	Δ_{total}	Δ_{leve}	B	B_{max}	β_T	β_x	T_m
1	27,03	23,50	4,52	5,38	12,0	12,0	1,21
2	35,00	29,75	5,30	6,04	9,50	9,50	1,42
3	40,00	34,00	5,60	6,38	14,0	14,0	1,60
4	11,50	9,78	4,37	4,98	17,0	17,0	1,05
5	20,25	17,21	4,85	5,52	10,4	10,4	1,34
6	24,00	20,40	5,37	6,12	17,0	17,0	1,40
7	40,00	34,00	5,60	6,38	14,0	14,0	1,60
8	2,29	1,95	2,23	4,57	20,2	20,2	1,26
9	6,60	5,35	2,80	2,80	16,5	16,5	0,96
10	3,85	2,94	2,50	2,50	15,0	15,0	0,96
11	11,50	9,00	3,50	3,50	15,0	15,0	1,16
12	24,90	21,40	4,26	4,26	15,9	15,9	1,48
13	31,00	27,00	4,74	4,74	17,8	17,8	1,45
14	44,00	37,40	5,29	5,29	12,0	12,0	1,78
15	73,70	62,65	6,29	6,29	12,0	12,0	1,82
16	23,70	19,50	4,75	4,75	12,0	12,0	1,30
17	46,50	39,50	5,29	5,29	12,0	12,0	1,80
18	38,00	32,00	5,38	5,40	13,0	15,0	1,75
19	60,50	51,43	5,75	5,75	13,8	13,8	1,98
20	17,34	14,74	4,10	4,10	15,0	15,0	1,20
21	13,50	11,48	3,94	3,94	15,0	15,0	1,20
22	14,00	11,82	4,09	4,16	17,9	17,0	1,28
23	8,03	7,20	3,60	3,80	18,0	18,0	1,00
24	11,51	9,01	4,06	4,06	17,0	17,0	1,09
25	19,30	16,10	4,75	4,75	14,0	19,0	1,35
26	22,98	19,98	4,57	4,65	20,0	20,0	1,10
27	12,40	10,60	3,80	4,00	17,9	17,9	1,12

Fonte: Autora (2022).

APÊNDICE C - CARACTERÍSTICAS DAS EMBARCAÇÕES ANALISADAS

Tabela 10 – Conjunto de dados (Parte II)

ID	Δ_{total}	Δ_{leve}	B	B_{max}	β_T	β_x	T_m
28	57,68	48,50	5,55	5,55	14,0	20,0	1,75
29	45,20	36,70	5,56	5,61	15,6	22,5	1,63
30	48,00	40,00	5,42	5,48	17,0	17,0	1,67
31	55,00	43,00	5,50	5,82	15,3	15,3	1,79
32	64,00	54,70	5,96	6,18	10,0	10,0	1,72
33	28,77	23,48	4,95	5,02	15,5	23,5	1,37
34	24,03	19,53	4,69	4,69	17,0	17,0	1,44
35	28,54	23,62	4,57	4,57	23,0	23,0	1,60
36	4,61	4,01	3,04	3,04	21,0	21,0	0,66
37	9,44	7,48	3,86	3,86	20,0	20,0	1,01
38	1,97	1,72	2,55	2,55	21,0	21,0	0,89
39	1,99	1,73	2,55	2,55	21,0	21,0	0,50
40	2,55	2,22	2,59	2,59	19,0	19,0	0,97
41	2,47	2,15	2,60	2,60	21,0	21,0	0,96
42	3,31	2,88	2,59	2,59	19,0	19,0	1,04
43	6,94	6,04	3,23	3,23	21,0	21,0	1,04
44	7,60	6,61	3,23	3,23	21,0	21,0	1,12
45	10,96	9,53	3,66	3,66	20,0	20,0	1,02
46	8,89	7,73	3,53	3,53	16,0	16,0	0,66
47	7,61	6,62	3,81	3,85	16,0	16,0	0,97
48	5,69	4,95	3,35	3,35	18,9	18,9	0,74
49	3,79	3,30	2,87	2,87	20,0	20,0	0,89
50	10,93	9,50	3,88	3,88	17,4	17,4	1,09
51	6,96	6,06	3,47	3,47	18,1	18,1	0,97
52	7,19	6,25	3,47	3,47	18,1	18,1	1,00
53	21,05	18,30	4,36	4,36	17,0	17,0	1,26
54	29,06	25,27	4,96	4,96	14,1	14,1	1,45

Fonte: Autora (2022).

APÊNDICE D - CARACTERÍSTICAS DAS EMBARCAÇÕES ANALISADAS

Tabela 11 – Conjunto de dados (Parte III)

ID	Z	LOA	L _{WL}	L _x	LCG	VCG	V _{cruzeiro}	V _{max}
1	2,79	17,63	15,57	7,60	6,55	0,56	32	35
2	3,27	19,02	16,35	8,17	7,36	1,63	25	30
3	3,45	21,86	18,80	9,40	8,46	1,72	25	30
4	2,69	12,98	11,16	5,58	5,02	1,34	30	38
5	2,99	16,57	14,25	7,12	6,41	1,49	27	35
6	3,31	18,80	16,16	8,08	7,27	1,65	30	39
7	3,45	22,82	19,62	9,81	8,83	1,72	25	30
8	3,04	8,480	8,407	3,59	2,95	0,13	32	45
9	1,86	10,07	9,920	4,96	4,46	0,93	35	41
10	1,66	8,240	7,080	3,54	3,18	0,83	35	40
11	2,33	11,88	10,10	5,05	4,54	1,16	31	40
12	2,84	14,92	12,80	6,40	5,76	1,42	30	35
13	3,16	17,27	13,99	6,99	6,29	1,58	28	34
14	3,52	20,67	16,97	8,48	7,63	1,76	33	37
15	4,19	26,84	21,42	10,7	9,63	2,09	34	38
16	2,75	17,39	14,95	7,47	6,73	1,37	32	39
17	3,52	20,54	16,97	8,48	7,63	1,76	33	37
18	3,78	21,31	18,32	9,16	8,24	1,89	32	34
19	3,83	23,25	19,57	9,78	8,80	1,91	28	32
20	3,98	14,50	12,47	6,23	5,61	1,99	37	50
21	3,82	12,90	11,09	5,54	4,99	1,91	33	45
22	4,39	13,28	11,42	5,71	5,14	2,19	28	32
23	3,65	10,67	9,176	4,58	4,12	1,82	31	35
24	3,55	12,19	10,48	5,24	4,71	1,77	26	30
25	3,77	14,46	12,43	6,21	5,59	1,88	26	31
26	3,07	16,68	14,35	7,17	6,45	1,53	31	34
27	3,65	12,00	10,32	5,16	4,64	1,82	28	32

Fonte: Autora (2022).

APÊNDICE E - CARACTERÍSTICAS DAS EMBARCAÇÕES ANALISADAS

Tabela 12 – Conjunto de dados (Parte IV)

ID	Z	LOA	L _{WL}	L _x	LCG	VCG	V _{cruzeiro}	V _{max}
28	5,38	24,63	21,18	10,59	9,53	2,69	27	34
29	5,48	21,62	17,28	8,64	7,77	2,74	27	33
30	2,75	21,30	17,04	8,52	7,66	1,37	24	29
31	2,77	22,94	19,72	9,86	8,87	1,38	27	30
32	6,19	23,71	20,70	10,35	9,31	3,09	24	28
33	5,96	17,88	15,37	7,68	6,92	2,98	28	32
34	4,67	16,78	14,43	7,21	6,49	2,33	29	32
35	2,28	17,98	15,46	7,73	6,95	1,14	40	48
36	1,90	8,99	7,733	3,86	3,48	0,95	24	30
37	2,66	12,39	10,66	5,33	4,79	1,33	23	31
38	1,40	7,75	6,665	3,33	2,99	0,70	19	39
39	1,40	7,01	6,029	3,01	2,71	0,70	25	39
40	1,42	7,16	6,158	3,07	2,77	0,71	25	35
41	1,42	7,50	6,450	3,22	2,90	0,71	22	20
42	1,42	8,10	6,966	3,48	3,13	0,71	28	36
43	1,77	9,83	8,454	4,22	3,80	0,88	26	36
44	1,77	10,59	9,107	4,55	4,09	0,88	28	36
45	2,00	11,51	9,902	4,95	4,45	1,00	30	45
46	1,67	10,28	7,671	3,83	3,45	0,83	25	36
47	1,82	11,35	9,764	4,88	4,39	0,91	26	38
48	1,98	10,33	8,884	4,44	3,99	0,99	28	40
49	1,69	9,780	8,411	4,20	3,78	0,84	26	33
50	2,29	12,25	10,53	5,26	4,74	1,14	25	33
51	2,05	11,34	9,752	4,87	4,38	1,02	27	33
52	2,05	11,34	9,752	4,87	4,38	1,02	26	32
53	2,57	15,82	13,60	6,80	6,12	1,28	24	27
54	2,93	18,08	15,54	7,77	6,99	1,46	24	29

Fonte: Autora (2022).

APÊNDICE F - RESISTÊNCIA ESTIMADA PELA PLANILHA DE (TWEEDIE, 2011)

Tabela 13 – Conjunto de dados (Parte V)

ID	V[nós]						
	15	20	25	30	35	40	45
1	21,16	26,41	29,46	31,72	34,51	38,29	43,11
2	28,43	36,42	41,19	44,35	47,83	52,47	58,47
3	29,60	38,82	45,65	51,19	56,63	62,97	70,64
4	12,32	14,97	16,71	18,93	21,96	25,81	30,41
5	17,43	22,16	25,28	28,20	31,90	36,62	42,37
6	19,92	26,27	31,09	35,50	40,59	46,78	54,17
7	28,71	37,77	44,82	50,97	57,07	64,00	72,18
8	3,63	4,43	5,46	6,85	8,60	10,71	13,15
9	7,15	8,51	9,48	10,64	12,18	14,14	16,51
10	5,24	5,51	5,86	6,58	7,65	9,03	10,70
11	14,23	15,98	16,27	16,91	18,30	20,41	23,16
12	28,98	34,76	35,73	35,61	36,35	38,32	41,43
13	34,07	42,18	44,39	44,66	45,67	48,05	51,80
14	38,35	48,97	54,49	56,90	59,06	62,40	67,27
15	52,93	70,56	81,78	88,93	93,95	99,28	106,1
16	20,51	25,83	29,02	31,52	34,58	38,64	43,75
17	41,43	52,88	58,55	60,61	62,29	65,18	69,66
18	29,11	37,71	43,82	48,49	53,11	58,68	65,55
19	47,85	62,42	71,39	76,46	79,95	84,05	89,69
20	18,21	22,12	23,91	25,52	27,92	31,29	35,59
21	15,46	18,14	19,21	20,62	22,88	26,04	30,01
22	16,02	19,18	20,67	22,46	25,15	28,79	33,32
23	10,44	11,72	12,64	14,18	16,46	19,40	22,97
24	13,75	15,87	16,92	18,53	21,02	24,33	28,42
25	20,64	24,79	26,24	27,86	30,58	34,46	39,41
26	22,23	27,98	31,19	33,49	36,33	40,16	45,03
27	15,81	17,99	18,60	19,70	21,70	24,56	28,19

Fonte: Autora (2022).

APÊNDICE G - RESISTÊNCIA ESTIMADA PELA PLANILHA DE (TWEEDIE, 2011)

Tabela 14 – Conjunto de dados (Parte VI)

ID	V[nós]						
	15	20	25	30	35	40	45
28	42,08	54,43	63,31	70,63	76,97	83,59	91,40
29	36,13	47,19	54,72	60,00	64,88	70,79	78,23
30	39,74	51,31	58,44	62,46	65,56	69,42	74,65
31	42,68	55,02	63,15	68,52	72,52	76,89	82,50
32	47,73	62,81	72,64	79,17	84,35	90,29	97,97
33	26,47	33,85	38,28	41,54	45,44	50,62	57,19
34	23,01	29,06	32,47	35,08	38,38	42,83	48,46
35	27,17	34,26	38,52	41,06	43,62	47,01	51,45
36	6,31	6,89	7,55	8,66	10,22	12,17	14,50
37	10,70	12,83	14,24	16,06	18,55	21,73	25,54
38	2,74	3,31	4,08	5,13	6,42	7,97	9,75
39	2,86	3,26	3,91	4,84	6,02	7,43	9,07
40	3,64	3,90	4,43	5,28	6,41	7,81	9,44
41	3,49	3,89	4,52	5,45	6,67	8,15	9,87
42	4,52	4,89	5,39	6,24	7,41	8,88	10,60
43	9,65	10,33	10,65	11,50	12,92	14,85	17,23
44	9,85	10,99	11,51	12,44	13,94	15,96	18,45
45	14,22	15,87	16,19	16,96	18,49	20,73	23,59
46	14,93	13,93	13,06	13,17	14,10	15,69	17,80
47	8,66	10,05	11,13	12,72	14,92	17,68	20,97
48	6,95	8,02	8,98	10,40	12,32	14,70	17,52
49	4,65	5,58	6,54	7,84	9,50	11,51	13,85
50	12,62	14,64	15,56	16,90	18,97	21,76	25,20
51	7,97	9,42	10,54	12,07	14,14	16,73	19,81
52	8,27	9,72	10,79	12,27	14,31	16,87	19,93
53	20,78	25,66	28,01	29,66	31,91	35,11	39,27
54	25,54	32,34	36,14	38,62	41,53	45,53	50,71

Fonte: Autora (2022).

APÊNDICE H - RESISTÊNCIA CALCULADA PELA REDE NEURAL M4

Tabela 15 – Resistência calculada por M4 (Parte I)

ID	V[nós]						
	15	20	25	30	35	40	45
1	25,79	30,26	34,72	39,19	41,50	43,54	45,06
2	33,06	37,53	42,00	46,47	50,94	54,02	54,36
3	36,12	40,59	45,06	49,53	54,00	58,47	61,49
4	11,78	13,82	15,86	17,90	19,95	21,99	24,03
5	19,45	23,92	28,39	30,69	32,73	34,49	34,82
6	21,24	25,71	30,18	33,15	35,19	37,23	39,28
7	35,52	39,99	44,46	48,93	53,40	57,87	61,27
8	0,00	2,00	4,00	6,00	8,00	10,00	12,00
9	7,06	9,10	11,14	13,18	15,22	17,26	19,30
10	4,60	6,64	8,68	10,72	12,76	14,80	16,33
11	13,71	15,75	17,80	19,84	21,88	23,92	25,96
12	26,54	31,01	34,92	36,96	39,00	41,05	43,09
13	31,60	36,07	40,54	44,26	46,30	48,34	50,38
14	42,84	47,31	51,78	56,25	60,72	65,19	68,18
15	70,07	74,54	79,01	83,48	87,95	92,42	96,89
16	23,34	27,81	32,28	35,45	37,49	39,53	40,78
17	45,48	49,95	54,42	58,89	63,36	67,83	71,38
18	35,45	39,92	44,39	48,86	53,33	57,46	59,50
19	57,91	62,38	66,85	71,32	75,79	80,26	84,73
20	18,96	22,82	24,86	26,90	28,94	30,99	33,03
21	15,91	17,95	19,99	22,04	24,08	26,12	28,16
22	15,70	17,74	19,78	21,82	23,86	25,91	27,95
23	8,07	10,11	12,15	14,19	16,24	18,28	20,32
24	12,66	14,70	16,74	18,78	20,82	22,86	24,90
25	20,66	24,51	26,55	28,59	30,63	32,67	34,72
26	22,87	27,34	30,43	32,47	34,51	36,56	38,60
27	13,73	15,77	17,82	19,86	21,90	23,94	25,98

Fonte: Autora (2022).

APÊNDICE I - RESISTÊNCIA CALCULADA PELA REDE NEURAL M4

Tabela 16 – Resistência calculada por M4 (Parte II)

ID \ V[nós]	15	20	25	30	35	40	45
28	54,24	58,71	63,18	67,65	72,12	76,59	81,06
29	43,76	48,23	52,70	57,17	61,64	66,05	68,09
30	46,68	51,15	55,62	60,09	64,56	69,03	72,46
31	51,86	56,33	60,80	65,27	69,74	74,21	78,68
32	60,49	64,96	69,43	73,90	78,37	82,84	86,06
33	28,38	32,85	37,32	39,89	41,93	43,97	46,01
34	23,96	28,43	32,66	34,70	36,74	38,78	40,82
35	27,94	32,41	36,88	39,15	41,19	43,23	45,27
36	3,42	5,46	7,50	9,54	11,58	13,63	15,67
37	9,06	11,10	13,14	15,18	17,22	19,26	21,30
38	0,85	2,86	4,86	6,86	8,86	10,86	12,86
39	0,95	2,95	4,95	6,95	8,95	10,95	12,95
40	1,85	3,89	5,94	7,98	10,02	12,06	14,10
41	1,24	3,24	5,24	7,24	9,27	11,31	13,36
42	2,70	4,74	6,79	8,83	10,87	12,91	14,95
43	6,34	8,38	10,42	12,46	14,51	16,55	18,59
44	7,09	9,13	11,17	13,21	15,25	17,30	19,34
45	11,51	13,55	15,60	17,64	19,68	21,72	23,76
46	10,41	12,45	14,49	16,53	18,57	20,61	22,65
47	7,85	9,89	11,94	13,98	16,02	18,06	20,10
48	4,98	7,02	9,06	11,11	13,15	15,19	17,23
49	2,52	4,56	6,60	8,64	10,68	12,72	14,76
50	11,86	13,90	15,94	17,98	20,03	22,07	24,11
51	6,66	8,70	10,74	12,78	14,82	16,86	18,90
52	6,98	9,02	11,06	13,10	15,14	17,18	19,22
53	21,75	26,22	28,98	31,02	33,06	35,11	37,15
54	28,31	32,78	37,25	41,72	44,06	46,10	48,14

Fonte: Autora (2022).

APÊNDICE J - RESISTÊNCIA CALCULADA PELA REDE NEURAL M12

Tabela 17 – Resistência calculada por M12 (Parte I)

ID \ V[nós]	15	20	25	30	35	40	45
1	22,89	28,44	32,11	35,79	39,46	43,13	45,42
2	30,04	36,78	42,51	46,19	49,86	53,53	57,10
3	32,17	38,91	45,65	50,24	53,91	57,58	61,25
4	11,99	15,73	17,70	19,03	20,35	21,67	22,99
5	17,53	21,45	25,12	28,79	32,46	36,05	37,30
6	20,14	25,27	28,94	32,62	36,29	39,19	40,44
7	30,48	37,22	43,96	48,90	52,57	56,24	59,92
8	3,30	4,18	5,06	5,94	6,82	7,70	8,58
9	4,64	8,08	9,40	10,72	12,04	13,37	14,69
10	4,58	5,90	7,22	8,54	9,86	11,19	12,51
11	12,23	15,97	18,04	19,36	20,69	22,01	23,33
12	27,03	31,09	34,83	37,15	38,47	39,79	41,11
13	32,48	38,93	42,67	45,80	47,13	48,45	49,77
14	39,36	46,10	52,83	58,49	62,16	65,84	69,09
15	63,61	70,86	78,11	84,90	91,64	97,94	102,99
16	20,17	25,12	28,79	32,46	36,13	39,80	41,56
17	42,21	48,94	55,68	61,93	65,60	69,28	71,93
18	30,95	37,69	44,43	49,57	53,24	56,91	60,46
19	53,69	60,43	67,17	73,91	80,65	85,32	87,63
20	17,90	21,58	25,25	27,57	28,88	30,20	31,52
21	14,77	18,51	21,08	22,40	23,72	25,04	26,37
22	16,03	19,77	21,74	23,07	24,39	25,71	27,03
23	10,58	11,90	13,23	14,55	15,87	17,19	18,52
24	13,95	16,76	18,08	19,40	20,73	22,05	23,37
25	22,88	26,46	27,78	29,10	30,43	31,75	33,07
26	23,63	27,47	31,21	34,17	35,49	36,81	38,13
27	15,96	17,74	19,06	20,38	21,70	23,03	24,35

Fonte: Autora (2022).

APÊNDICE K - RESISTÊNCIA CALCULADA PELA REDE NEURAL M12

Tabela 18 – Resistência calculada por M12 (Parte II)

ID \ V[nós]	15	20	25	30	35	40	45
28	49,47	56,21	62,95	69,69	76,43	82,84	84,81
29	43,83	50,63	56,96	61,35	65,53	66,85	68,18
30	44,91	51,72	58,52	64,59	68,33	70,63	71,78
31	47,31	54,05	60,79	67,53	73,05	76,73	80,40
32	54,90	62,15	69,40	76,14	82,88	89,62	92,45
33	29,73	34,42	38,81	41,10	42,42	43,75	45,07
34	23,77	29,00	32,68	36,42	37,85	39,18	40,50
35	28,48	33,51	37,25	40,99	42,37	43,52	44,68
36	5,49	6,20	6,91	7,62	8,77	9,92	11,07
37	10,86	13,36	14,68	16,00	17,32	18,65	19,97
38	3,65	4,36	5,07	5,78	6,49	7,20	7,91
39	3,30	4,01	4,72	5,43	6,14	6,84	7,55
40	3,90	4,61	5,32	6,03	6,74	7,44	8,15
41	3,72	4,43	5,14	5,84	6,55	7,26	7,97
42	4,65	5,36	6,07	6,78	7,58	8,73	9,89
43	7,49	8,64	9,79	10,94	12,09	13,24	14,39
44	8,98	10,30	11,47	12,63	13,78	14,93	16,08
45	13,69	15,01	16,33	17,65	18,98	20,24	21,39
46	12,44	13,76	15,08	16,41	17,73	19,05	20,37
47	7,82	11,42	12,74	14,07	15,39	16,71	18,03
48	6,79	8,11	9,43	10,76	12,08	13,40	14,72
49	4,36	5,24	6,52	7,84	9,17	10,49	11,81
50	11,89	15,63	16,97	18,30	19,62	20,94	22,26
51	7,51	10,10	11,43	12,75	14,07	15,39	16,72
52	7,83	10,41	11,74	13,06	14,38	15,70	17,03
53	20,53	24,27	28,01	31,75	33,32	34,64	35,97
54	25,78	32,24	35,91	39,59	43,26	46,54	47,79

Fonte: Autora (2022).

APÊNDICE L - RESISTÊNCIA CALCULADA PELA REDE NEURAL M1

Tabela 19 – Resistência calculada por M1 (Parte I)

ID	V[nós]						
	15	20	25	30	35	40	45
1	33,16	34,77	36,38	38,00	39,61	41,22	42,84
2	42,10	43,71	45,32	46,94	48,55	50,16	51,77
3	45,77	47,38	48,99	50,61	52,22	53,83	55,45
4	15,10	16,71	18,33	19,94	21,55	23,17	24,78
5	27,39	29,01	30,62	32,23	33,85	35,46	37,07
6	28,98	30,59	32,21	33,82	35,43	37,05	38,66
7	46,10	47,71	49,33	50,94	52,55	54,17	55,78
8	8,02	7,36	7,35	8,96	10,58	12,19	13,80
9	9,05	10,66	12,27	13,89	15,50	17,11	18,73
10	5,92	7,53	9,15	10,76	12,37	13,99	15,60
11	14,59	16,21	17,82	19,43	21,05	22,66	24,27
12	27,70	29,32	30,93	32,54	34,15	35,77	37,38
13	33,08	34,70	36,31	37,92	39,54	41,15	42,76
14	48,76	50,38	51,99	53,60	55,22	56,83	58,44
15	77,63	79,25	80,86	82,47	84,09	85,70	87,31
16	29,54	31,15	32,77	34,38	35,99	37,60	39,22
17	50,99	52,60	54,21	55,83	57,44	59,05	60,67
18	43,34	44,95	46,57	48,18	49,79	51,41	53,02
19	64,00	65,61	67,23	68,84	70,45	72,06	73,68
20	21,13	22,75	24,36	25,97	27,59	29,20	30,81
21	17,06	18,67	20,28	21,90	23,51	25,12	26,74
22	16,63	18,24	19,85	21,47	23,08	24,69	26,30
23	10,01	11,62	13,24	14,85	16,46	18,08	19,69
24	14,24	15,86	17,47	19,08	20,70	22,31	23,92
25	23,07	24,69	26,30	27,91	29,53	31,14	32,75
26	25,07	26,68	28,29	29,91	31,52	33,13	34,74
27	14,53	16,14	17,76	19,37	20,98	22,60	24,21

Fonte: Autora (2022).

APÊNDICE M - RESISTÊNCIA CALCULADA PELA REDE NEURAL M1

Tabela 20 – Resistência calculada por M1 (Parte II)

ID	V[nós]						
	15	20	25	30	35	40	45
28	61,14	62,75	64,36	65,97	67,59	69,20	70,81
29	47,68	49,29	50,90	52,52	54,13	55,74	57,35
30	50,27	51,89	53,50	55,11	56,73	58,34	59,95
31	58,55	60,17	61,78	63,39	65,01	66,62	68,23
32	69,52	71,13	72,75	74,36	75,97	77,59	79,20
33	31,81	33,43	35,04	36,65	38,27	39,88	41,49
34	27,39	29,00	30,62	32,23	33,84	35,46	37,07
35	29,09	30,70	32,32	33,93	35,54	37,16	38,77
36	7,34	6,68	7,80	9,41	11,03	12,64	14,25
37	11,02	12,63	14,24	15,86	17,47	19,08	20,70
38	5,72	5,06	4,70	6,31	7,93	9,54	11,15
39	5,43	4,77	4,46	6,07	7,69	9,30	10,91
40	5,47	4,81	5,92	7,53	9,14	10,76	12,37
41	5,80	5,14	5,09	6,70	8,32	9,93	11,54
42	6,06	5,40	6,92	8,53	10,15	11,76	13,37
43	8,49	8,67	10,28	11,89	13,51	15,12	16,73
44	8,99	9,52	11,13	12,75	14,36	15,97	17,59
45	11,94	13,55	15,16	16,77	18,39	20,00	21,61
46	10,87	12,49	14,10	15,71	17,32	18,94	20,55
47	10,78	12,40	14,01	15,62	17,23	18,85	20,46
48	8,19	8,73	10,34	11,96	13,57	15,18	16,80
49	7,19	6,53	7,67	9,28	10,90	12,51	14,12
50	13,45	15,07	16,68	18,29	19,91	21,52	23,13
51	9,03	10,64	12,25	13,87	15,48	17,09	18,71
52	9,23	10,85	12,46	14,07	15,69	17,30	18,91
53	24,18	25,80	27,41	29,02	30,64	32,25	33,86
54	33,76	35,37	36,98	38,60	40,21	41,82	43,44

Fonte: Autora (2022).