UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

NATTANE LUÍZA DA COSTA

Uso e estabilidade de seletores de variáveis baseados nos pesos de conexão de redes neurais artificiais

Goiânia
2021

Processo: 23070.012654/2021-91 Documento: 1999331

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES

E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico

[ ] Dissertação [x] Tese

2. Nome completo do autor

Nattane Luíza da Costa

3. Título do trabalho

Uso e estabilidade de seletores de variáveis baseados nos pesos de conexão de redes neurais artificiais

4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concordo com a liberação total do documento [x] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:
a) consulta ao(à) autor(a) e ao(à) orientador(a);
b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.
O documento não será disponibilizado durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Rommel Melgaço Barbosa**, **Professor do Magistério Superior**, em 13/04/2021, às 15:52, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **NATTANE LUÍZA DA COSTA**, **Discente**, em 14/04/2021, às

Processo: 23070.012654/2021-91 Documento: 1999331

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1999331** e o código CRC **7B391C1A**.

Referência: Processo nº 23070.012654/2021-91 SEI nº 1999331

NATTANE LUÍZA DA COSTA

Uso e estabilidade de seletores de variáveis baseados nos pesos de conexão de redes neurais artificiais

Tese apresentada ao Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Ciência da Computação.

Orientador: Prof. Rommel Melgaço Barbosa

Goiânia
2021

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Costa, Nattane Luíza da
Uso e estabilidade de seletores de variáveis baseados nos pesos de conexão de redes neurais artificiais [manuscrito] / Nattane Luíza da Costa. - 2021.
CLXXV, 175 f.

Orientador: Prof. Dr. Rommel Melgaço Barbosa.
Tese (Doutorado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Cidade de Goiás, 2021.

Inclui siglas, símbolos, algoritmos, lista de figuras, lista de tabelas.

1. redes neurais artificiais. 2. seleção de variáveis. 3. pesos de conexão. 4. estabilidade. 5. ranking de importância. I. Barbosa, Rommel Melgaço, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE TESE

Ata nº **08/2021** da sessão de Defesa de Tese de **Nattane Luíza da Costa**, que confere o título de Doutora em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos dezenove dias do mês março de dois mil e vinte e um, a partir das catorze horas, via sistema de webconferência da RNP, realizou-se a sessão pública de Defesa de Tese intitulada **"Uso e estabilidade de seletores de variáveis baseados nos pesos de conexão de redes neurais artificiais"**. Os trabalhos foram instalados pelo Orientador, Professor Doutor Rommel Melgaço Barbosa (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professora Doutora Isis Didier Lins (UFPE), membra titular externa; Professor Doutor Márcio Dias de Lima (IFG), membro titular externo; Professor Doutor Plínio de Sá Leitão Júnior (INF/UFG), membro titular interno; Professor Doutor Ronaldo Martins da Costa (INF/UFG), membro titular interno. A realização da banca ocorreu por meio de videoconferência, em atendimento à recomendação de suspensão das atividades presenciais na UFG emitida pelo Comitê UFG para o Gerenciamento da Crise COVID-19, bem como à recomendação de isolamento social da Organização Mundial de Saúde e do Ministério da Saúde para enfrentamento da emergência de saúde pública decorrente do novo coronavírus. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Tese, tendo sido a candidata **aprovada** pelos seus membros. Proclamados os resultados pelo Professor Doutor Rommel Melgaço Barbosa, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos dezenove dias do mês março de dois mil e vinte e um.

TÍTULO SUGERIDO PELA BANCA

Documento assinado eletronicamente por **Rommel Melgaço Barbosa**, **Professor do Magistério Superior**, em 19/03/2021, às 17:24, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Márcio Dias de Lima**, **Usuário Externo**, em 19/03/2021, às 17:24, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Ronaldo Martins Da Costa**, **Vice-Coordenador de Pós-graduação**, em 19/03/2021, às 17:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Plinio De Sa Leitão Junior**, **Professor do Magistério Superior**, em 19/03/2021, às 17:26, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do

Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Isis Didier Lins**, **Usuário Externo**, em 19/03/2021, às 17:27, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **NATTANE LUÍZA DA COSTA**, **Discente**, em 19/03/2021, às 18:22, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1938321** e o código CRC **6F60BCB8**.

Referência: Processo nº 23070.012654/2021-91 SEI nº 1938321

Nattane Luíza da Costa

Graduou-se em Tecnologia em Redes de Computadores pela Universidade Estadual de Goiás (2014). Possui mestrado em Ciência da Computação pela Universidade Federal de Goiás (2016). Durante o Mestrado, aprofundou seus estudos em mineração de dados para problemas de classificação, com aplicação para o reconhecimento geográfico e de variedades de alguns vinhos *Vitis Vinífera* da América do Sul. Atualmente é Professora no Instituto Federal de Educação, Ciência e Tecnologia Goiano, atuando no campus Urutaí e tem interesse em pesquisas sobre mineração de dados, aprendizado de máquina e seleção de variáveis.

Aos meus pais.

Agradecimentos

A Deus, inteligência suprema causa primária de todas as coisas, pela dádiva da vida e por ter me concedido a oportunidade de caminhar ao lado de pessoas amadas.

Agradeço ao meu orientador professor Dr. Rommel Melgaço Barbosa, por sua orientação, disponibilidade, acessibilidade, conhecimento, pelos incentivos e confiança.

Agradeço aos meu pais, Barsanulfo Zaruh da Costa e Adriana Luíza G. da Costa, por me proporcionarem todo o carinho, amor, apoio incondicional, suporte físico e emocional ao longo de toda a minha jornada acadêmica. Amo vocês. Ao meu irmão Maxwell Severo da Costa, juntamente com minha cunhada Hélen e meus sobrinhos Matheus e Luíza, por serem o apoio sempre que precisei, pelas conversas e momentos em família.

Ao meu esposo Renato Gomes Borges Júnior, pelo companheirismo, amor, amizade, paciência e humor ao me motivar sempre que algum experimento não saía como o planejado. Obrigada pelas conversas e correções que contribuíram para o desenvolvimento desta tese.

Agradeço aos meus amigos de pós-graduação Diego, Márcio, Guilherme e Camila por todo o apoio e momentos de distração, e de modo especial ao Diego e Márcio pelas parcerias e amizade, que muito contribuíram no desenvolvimento deste trabalho.

Agradeço a minha amiga de infância Andrezza Arantes Castro, companheira de pós-graduação, colega de quarto no início de nossa trajetória, e hoje colega de profissão. Muito obrigada por todos os momentos juntas, pelas conversas e discussão sobre ciência, pela paciência e pelos momentos de distração.

Agradeço aos colegas professores do Núcleo de Informática do Instituto Federal Goiano pelo apoio e incentivo. Agradeço ao Instituto Federal Goiano Campus Urutaí pela concessão do afastamento, que mesmo por um breve período de tempo, possibilitou me dedicar integralmente ao desenvolvimento desta tese.

Agradeço aos meus padrinhos José Chaves, Edilma e Glória por todo o apoio desde o meu primeiro dia de mestrado na cidade de Goiânia.

Agradeço aos servidores técnicos administrativos do Instituto de Informática da Universidade Federal de Goiás, Mariana e Mirian, pela atenção e dedicação em seus trabalhos realizados para o bom andamento do doutorado.

Agradeço à banca examinadora, pelas considerações que muito contribuíram para a versão final deste texto.

A todos que direta ou indiretamente contribuíram para a conclusão desta tese, muito obrigada!

Theory without data is fantasy; but data without theory is chaos.

Edward E. Lawler, 1971,

.

Resumo

Costa, Nattane Luíza da. **Uso e estabilidade de seletores de variáveis baseados nos pesos de conexão de redes neurais artificiais**. Goiânia, 2021. 175p. Tese de Doutorado, Instituto de Informática, Universidade Federal de Goiás.

Redes Neurais Artificiais (RNA) são modelos de aprendizado de máquina usados para a modelagem de problemas em vários campos de pesquisa. Contudo, as RNAs são frequentemente consideradas como caixas pretas, o que significa que esses modelos não podem ser interpretados, pois não fornecem informações explicativas. Os seletores de variáveis com base nos pesos de conexão (*Weight Based feature selectors* - WBFS) foram propostos para de extrair conhecimento das RNAs por meio de um ranking de importância das variáveis de entrada em relação à saída da rede. Grande parte das aplicações que utilizam esses algoritmos se baseiam em apenas um modelo de RNA. No entanto, existe uma variação nos valores dos pesos de conexão de RNAs treinadas em diferentes momentos de inicialização dos pesos e em razão do treinamento da rede, ocorrendo assim, uma variação no ranking de importância de um mesmo conjunto de dados. Neste contexto, esta tese apresenta um estudo sobre os WBFS. Primeiro, propomos uma nova abordagem de votação para avaliar a estabilidade dos WBFS, ou seja, a variação obtida ao gerar rankings de importância por meio de muitos modelos de RNA. Em seguida, avaliamos a estabilidade dos WBFS com base em *multilayer perceptron* (MLP) e *extreme learning machines* (ELM). Ademais, propomos uma melhoria nos WBFS de Garson, Olden e de Yoon, combinando-os com o seletor de variáveis *ReliefF*, denominados FSGR, FSOR e FSYR. Os experimentos foram realizados com base em 28 arquiteturas MLP, 16 arquiteturas ELM e 16 conjuntos de dados do repositório UCI. Os resultados mostram que há uma diferença significativa na estabilidade WBFS dependendo dos parâmetros de treinamento das RNAs e a depender do WBFS utilizado. Os algoritmos propostos se demonstraram mais eficazes do que os algoritmos clássicos. Até onde sabemos, este estudo foi a primeira tentativa de medir a estabilidade dos WBFS, de investigar os efeitos dos parâmetros da RNA na estabilidade do WBFS e o primeiro a propor uma combinação de WBFS com outro seletor de variáveis. Além disso, os resultados fornecem informações sobre os benefícios e as limitações dos WBFS e representam um ponto de partida para melhorar a estabilidade desses algoritmos.

Palavras–chave

redes neurais artificiais, seleção de variáveis, pesos de conexão, estabilidade, ranking de importância.

Abstract

Costa, Nattane Luíza da. **The use and stability of feature selectors based on artificial neural network connection weights**. Goiânia, 2021. 175p. PhD. Thesis Relatório de Graduação. Instituto de Informática, Universidade Federal de Goiás.

Artificial Neural Networks (ANN) are machine learning models used to solve problems in several research fields. Although, ANNs are often considered "black boxes", which means that these models cannot be interpreted, as they do not provide explanatory information. Connection Weight Based Feature Selectors (WBFS) have been proposed to extract knowledge from ANNs. Most of studies that have been using these algorithms are based on just one ANN model. However, there are variations in the ANN connection weight values due to the initialization and training, and consequently, leading to variations in the importance ranking generated by a WBFS. In this context, this thesis presents a study about the WBFS. First, a new voting approach is proposed to assess the stability of the WBFS, i.e, the variation in the result of the WBFS. Then, we evaluated the stability of the algorithms based on multilayer perceptron (MLP) and extreme learning machines (ELM). Furthermore, an improvement is proposed in the algorithms of Garson, Olden, and Yoon, combining them with the feature selector ReliefF. The new algorithms are called FSGR, FSOR, and FSYR. The experiments were performed based on 28 MLP architectures, 16 ELM architectures, and 16 data sets from the UCI Machine Learning Repository. The results show that there is a significant difference in WBFS stability depending on the training parameters of the ANNs and depending on the WBFS used. In addition, the proposed algorithms proved to be more effective than the classic algorithms. As far as we know, this study was the first attempt to measure the stability of WBFS, to investigate the effects of different ANN training parameters on the stability of WBFS, and the first to propose a combination of WBFS with another feature selector. Besides, the results provide information about the benefits and limitations of WBFS and represent a starting point for improving the stability of these algorithms.

Keywords

artificial neural network, feature selection, connection weights, stability, importance rankings

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Siglas

BP	*Backpropagation*
BRNN	*Bayesian Regularization Neural Network*
CD	*Critical difference*
DTM	*Document term matrix*
ELM	*Extreme Learning Machines*
FSGR	*Feature selector based on Garson's and ReliefF algorithm*
FSOR	*Feature selector based on Olden's and ReliefF algorithm*
FSYR	*Feature selector based on Yoon's and ReliefF algorithm*
FN	Falso Negativo
FP	Falso Positivo
HN	Quatidade de neurônios ocultos - *Hidden Neuros*
IBP	*Incremental Backpropagation*
LM	*Levenberg Marquart*
MLP	*Multilayer Perceptron*
NO	Número de neurônios ocultos
NRMSE	Raiz quadrada do erro médio normalizado
PSO	*Partial swarm optmization*
RMSE	Raiz quadrada do erro médio
RNA	Redes Neurais Artificiais
RPROP	*Backpropagation* resiliente
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
WBFS	*Weight based feature selection*
WoS	*Web of Science*

Lista de Símbolos

$\mathbb{R}^m$	Espaço m-dimensional
$\mathbf{u}$	vetor
$E(W)$	erro da rede neural com os pesos $\mathbf{w}$
λ	*Weight decay*
$median()$	Função que retorna a mediana do vetor $\mathbf{x}$
$max()$	Função que retorna o valor máximo do vetor $\mathbf{x}$
$min()$	Função que retorna o valor mínimo do vetor $\mathbf{x}$
S_r	Correlação de Spearman
$\langle \mathbf{x}, \mathbf{y} \rangle$	produto interno entre os vetores $\mathbf{x}$ e $\mathbf{y}$

CAPÍTULO 1

Introdução

Redes neurais artificiais (RNA) são algoritmos de aprendizagem de máquina baseados no comportamento biológico do cérebro humano capazes de generalizar problemas complexos. A família de algoritmos de RNA são amplamente utilizados em problemas relacionados com a inteligência artificial [2] e com outros campos de pesquisa como a análise de qualidade da água [142], processamento de imagem [161], ciência de alimentos [23, 19], saúde [70, 128], entre outros.

As RNAs são compostas por várias unidades computacionais, denominadas neurônios artificiais, distribuídos em pelo menos três camadas. A primeira camada serve de entrada para as variáveis que descrevem os dados, chamada de camada de entrada. Essa camada é diretamente conectada à uma ou mais camadas ocultas responsáveis por computar os dados. Por fim, a última camada, denominada camada de saída, é responsável por realizar os cálculos para determinar a saída da rede. A conexão entre os neurônios de cada camada é feita por meio de pesos de conexão que são utilizados para computar a saída de cada neurônio.

Dada essa arquitetura, os dados são propagados entre os neurônios, partindo da camada de entrada, passando pelas camadas ocultas até a camada de saída, em um processo chamado *forward propagation*. Em seguida, a saída resultante da rede é comparada com a saída esperada. O erro obtido é então retro-propagado pela rede para que seja feito um ajuste nos pesos de conexão que ligam os neurônios, de forma a minimizar este erro. Esse processo é chamado de *backward propagation*. Ao final do processo de treinamento, o resultado obtido é uma matriz de pesos de entrada e de saída que são utilizados para determinar a classe de uma nova amostra.

O aprendizado de uma RNA é comumente caracterizado como uma “caixa preta” [53, 38]. Essa caracterização se dá devido ao fato de a modelagem de um problema por meio de RNA possuir muitas elementos com relações complexas entre si que são específicas para os dados de um contexto particular utilizados para a modelagem, o que torna difícil descrever todo o processo de classificação determinado pela rede [45]. O modelo é representado por uma matriz de pesos, que por si só não representam conhecimento útil, nem descrevem o porquê de uma amostra pertencer a uma classe ou à

outra.

Nesse contexto, alguns métodos para estimar a contribuição relativa das variáveis de entrada foram propostas ao longo do tempo na tentativa de tornar um modelo de rede neural mais compreensível. Existem três categorias principais desses métodos. A primeira categoria diz respeito aos métodos baseados na matriz de pesos de uma RNA. Esses métodos usam os valores dos pesos de conexão para calcular a importância de uma variável de entrada para a saída da RNA.

A segunda categoria descreve os métodos de análise de sensitividade. Esses métodos funcionam a partir da análise dos efeitos provocados na saída da rede por meio de uma variação dos valores de cada variável de entrada sobre uma faixa de valores totais ou parciais, mantendo todas as outras variáveis de entrada constantes em um percentil especificado. Outro exemplo de análise de sensitividade é a análise das derivadas parciais da saída em relação a uma entrada específica da RNA.

Por último, a terceira categoria descreve os algoritmos que geram regras a partir de um modelo de RNA. Esses algoritmos resultam em regras do tipo "se-então-senão", que são facilmente compreensíveis, mas não ilustram a importância relativa das variáveis de entrada de maneira individual. Além disso, esses algoritmos podem resultar em um extenso conjunto de regras inflexíveis [165].

A primeira categoria, referida neste trabalho como seletores de variáveis baseados nos pesos de conexão de uma RNA (*Weight based feature selectors - WBFS*), são o foco desta tese de doutorado. Essa categoria foi escolhida pois os WBFS funcionam como um método de seleção de variáveis do tipo filtro, classificando as variáveis em uma ordem de importância. Os seletores de variáveis do tipo filtro são geralmente usados como uma etapa de pré-processamento na mineração de dados. Frequentemente, as variáveis com os maiores valores de importância são selecionados e aplicados à um classificador [16]. Em contrapartida, os WBFS são comumente usados após a fase de treinamento do modelo e o ranking de importância resultante é usado para discutir e concluir sobre o tópico que foi estudado.

Existem cinco WBFS que têm sido utilizados em diferentes aplicações: o algoritmo de Garson [44, 46, 164], o algoritmo de Olden [102, 106, 122], o algoritmo de Yoon [76, 77, 145, 144], algoritmo de Tsaur [146, 144], e o algoritmo de Howes e Crook [52, 109, 130].

Nos estudos encontrados foram identificadas duas formas principais de uso de um WBFS. A primeira considera apenas um modelo de RNA para aplicar o WBFS. Essa categoria pode ainda se dividir em duas especializações i) a modelagem de apenas uma arquitetura de RNA e uso do WBFS, e ii) a modelagem de diferentes arquiteturas de RNAs e uso do WBFS apenas no modelo com o melhor desempenho. Em ambas especializações o resultado é o mesmo: um vetor ordenado com os valores de importância de cada variável

(ranking de importância).

A segunda categoria considera um conjunto de RNAs, em que o WBFS é aplicado em todos os modelos. O resultado final é então calculado a partir da média dos valores de importância obtido em cada ranking. Os estudos que utilizaram essa abordagem relataram uma variação nos valores de importância gerados pelo WBFS a partir de diferentes modelos de RNAs [27, 35, 97]. Essa variação ocorre devido à inicialização aleatória dos pesos de conexão e ao processo de treinamento, resultando em diferentes matrizes de pesos e, consequentemente, resultando em diferentes valores de importância para cada variável. Isso ocorre devido ao fato de que o ranking de importância é calculado com base na matriz de pesos resultante da RNA. O Capítulo 3 descreve o estudo que deu origem a definição dessas duas categorias do uso dos WBFS.

Nesse sentido, nossa primeira hipótese (**H1**) afirma que os valores de importância obtidos por um WBFS com base em um conjunto de RNAs resultam em diferentes rankings de importância, nos quais uma mesma variável pode ser classificada em diferentes posições de importância. Ou seja, uma dada variável pode ser classificada como uma das mais importantes com base em um modelo de RNA, e em outro modelo de RNA construído com base nos mesmos dados de entrada e treinada em um diferente momento, a mesma variável pode ser classificada como uma de menor importância.

Acreditamos que a instabilidade observada por diversos autores [27, 35, 97] não deve ser avaliada a partir do valor médio de importância e o desvio padrão, visto que os resultados podem ser não confiáveis, aleatórios e irreproduzíveis. A estabilidade dos resultados de algoritmos de seleção de variáveis indica seu poder de reprodutibilidade, que é tão importante quanto o desempenho e reprodutibilidade de um algoritmo de classificação [65]. No caso de nossa hipótese ser corroborada, o uso dos valores médios de importância se torna inviável visto que uma mesma variável pode receber valores de importância relacionados a praticamente todas as posições de importância.

Dessa maneira, no Capítulo 4 nós propomos uma nova metodologia para avaliar a estabilidade de métodos WBFS com base em um conjunto de RNAs. Essa metodologia é baseada em posições de importância que são calculadas por meio de uma abordagem de votação. Essa proposta se apresentou mais confiável do que o uso do valor médio de importância, pois explora se todos os rankings resultam em ordens de importância semelhantes. Três métricas foram estabelecidas para a avaliação da estabilidade de um WBFS: confiança, confiança mínima e a quantidade de posições votadas. Os algoritmos de Garson, Olden, Yoon, Tsaur e de Howes e Crook foram usados para avaliar a metodologia proposta em conjuntos de dados simulados e em conjuntos de dados reais do UCI *Machine Leaning Repository*. Os resultados desse capítulo foram publicados no *Journal Expert Systems with Applications* [21].

Outras hipóteses surgiram a partir dos resultados obtidos na validação da meto-

dologia da abordagem de votação. A segunda hipótese desta tese (**H2**) afirma que existe diferença na estabilidade dos rankings de importância obtidos por meio de WBFS com base em RNAs com diferentes parâmetros de treinamento. Sabe-se que os parâmetros de treinamento de uma RNA influenciam na capacidade de predição do modelo. No entanto, não há estudos que avaliam o impacto desses parâmetros no resultado de um WBFS.

A terceira hipótese (**H3**) afirma que existe uma diferença na estabilidade dos rankings de importância obtidos por meio de um WBFS com base em RNAs do tipo *Multilayer Perceptron* (MLP) e RNAs do tipo *Extreme Learning Machines* (ELM). Até o presente momento não há estudos na literatura que avaliam a instabilidade em MLP, ELM, ou qualquer outro tipo de RNA. O MLP foi escolhido por ser um dos modelos de RNA mais utilizados juntamente com o algoritmo de *backpropagation* [117]. Esse tipo de RNA pode consumir uma enorme quantidade de tempo computacional devido à atualização dos pesos de conexão e otimização de parâmetros da rede. Em contraste, ELM é um modelo de RNA que não requer tal tempo computacional porque seus pesos de conexão que ligam a camada de entrada com a camada oculta são atribuídos aleatoriamente, enquanto apenas os pesos de conexão da camada de saída são calculados uma única vez [59]. Esse algoritmo tem sido usado com sucesso em muitos campos de pesquisa, como engenharia de recursos hídricos [153], classificação de alimentos [23], reconstrução de imagens [47] e reconhecimento de emoções [55].

A quarta hipótese (**H4**) desta tese afirma que uma ponderação do cálculo de um WBFS melhora a estabilidade dos métodos, ou seja, associar uma métrica de importância ao cálculo dos WBFS pode aumentar a estabilidade. De maneira a testar essa hipótese nós propomos uma melhoria dos algoritmos de Garson, Olden de Yoon por meio do algoritmo *ReliefF*. Apenas esses três algoritmos foram escolhidos para essa fase do estudo, pois, conforme será demonstrado no Capítulo 4, eles possuem desempenho equivalentes entre si e desempenho superior aos algoritmos de Tsaur e de Howes e Crook. Os novos algoritmos são denominados *feature selector based on Garson's and ReliefF algorithm* (FSGR), *feature selector based on Olden 's and ReliefF algorithm* (FSOR) e *feature selector based on Yoon 's and ReliefF algorithm* (FSYR). O algoritmo ReliefF foi escolhido devido à sua simplicidade e eficácia [139]. Além disso, esse algoritmo tem sido usado com sucesso em diferentes campos de pesquisa como bioinformática [140], ciência de alimentos [19, 151] e avaliação de risco de crédito [8].

A quinta hipótese (**H5**) afirma que existe diferença nas medidas de estabilidade obtidas por diferentes WBFS. Até o presente momento, não há estudos na literatura que avaliam a estabilidade de diferentes WBFS. Para testar essa hipótese nós comparamos os algoritmos de Garson, Olden, e de Yoon, e os algoritmos propostos FSGR, FSOR e FSYR em conjunto com as hipóteses **H2** e **H3**.

Por fim, a sexta hipótese (**H6**) desta tese afirma que existe uma diferença no

ranking de importância final obtido pela abordagem de votação a depender do WBFS e da rede neural utilizada. Para atestar essa hipótese nós avaliamos os rankings de importância finais obtidos pelos algoritmos de Garson, Olden, Yoon, FSGR, FSOR e FSYR baseados em RNAs do tipo MLP e ELM utilizando o coeficiente de correlação de postos de Spearman (*Spearman's rank correlation coefficient*).

A viabilidade das hipóteses **H2**, **H3**, **H4**, **H5** e **H6** são atestadas no capítulo 5 por meio de uma série de experimentos em diferentes cenários de parâmetros de treinamento dos algoritmos MLP e ELM, com base em 16 conjuntos de dados do UCI *Machine Leaning Repository*. Adicionalmente, nós avaliamos qual o número médio de RNAs que são necessárias para que os WBFS atingissem sua possível estabilidade.

Com base em nossa ampla investigação e resultados experimentais significativos, a viabilidade das hipóteses **H1**, **H2**, **H3**, **H4**, **H5** e **H6** foi constatada. As hipóteses **H2** e **H6** foram ajustadas conforme os resultados obtidos para cenários específicos. Até onde se sabe, a estabilidade dos WBFS nunca foi estudada. Os resultados obtidos fornecem uma visão sobre a aplicabilidade, benefícios e limitações de algoritmos do tipo WBFS. Por fim, os resultados obtidos nesta tese foram publicados em revistas científicas ou estão em processo de revisão por pares.

1.1 Publicações

Durante o período de curso do doutorado, dez estudos foram publicados em revistas científicas internacionais bem conceituadas. Algumas dessas aplicações foram iniciadas no período do mestrado e aprimoradas no doutorado. Sete destes trabalhos apresentam aplicações de técnicas de aprendizado de máquina e mineração de dados para a análise e/ou classificação de alimentos e outras substâncias; um trabalho apresenta uma melhoria de uma variação do algoritmo Máquinas Vetores de Suporte; um estudo apresenta uma revisão bibliográfica sobre o uso de aprendizado de máquina e quimiometria para a análise de amostras de cervejas; e por fim, um artigo é diretamente relacionado aos resultados apresentados nesta tese de doutorado.

Os artigos publicados, em ordem cronológica foram:

1. Geographical recognition of Syrah wines by combining feature selection with Extreme Learning Machine (2018). *Measurement*, 120, 92-99 [23]. Fator de Impacto 3,364.
2. Advanced data mining approaches in the assessment of urinary concentrations of bisphenols, chlorophenols, parabens and benzophenones in Brazilian children and their association to DNA damage (2018). *Environment international*, 116, 269-277 [113]. Fator de Impacto 7,577.

3. Improvements on least squares twin multi-class classification support vector machine (2018). *Neurocomputing*, 313, 196-205 [26]. Fator de Impacto 4,438.
4. Geographical classification of Tannat wines based on support vector machines and feature selection (2018). *Beverages*, 4 (4), 97 [17].
5. The use of data mining to classify Carménère and Merlot wines from Chile (2019). *Expert Systems*, 36 (2), e12361 [22]. Fator de Impacto: 1,546.
6. Using Support Vector Machines and neural networks to classify Merlot wines from South America (2019). *Information Processing in Agriculture*, 6 (2), 265-278 [19]. Fator de Impacto 1,984.
7. Finding the most important sensory descriptors to differentiate some Vitis vinifera L. South American wines using support vector machines (2019). *European Food Research and Technology*, 245 (6), 1207-1228 [18]. Fator de Impacto 2,366.
8. Characterization of Cabernet Sauvignon wines from California: determination of origin based on ICP-MS analysis and machine learning techniques (2020). *European Food Research and Technology*, 1-13 [24]. Fator de Impacto 2,366.
9. A review on the application of chemometrics and machine learning algorithms to beer evaluation (2020). *Food Analytical Methods*, p. 1-20 [20]. Fator de Impacto 2,667.
10. Evaluation of feature selection methods based on artificial neural network weights (2021). *Expert Systems with Applications*, v. 168, p. 114312. [21]. Fator de Impacto 5,452.

Além dessas publicações, os seguintes estudos estão submetidos e em processo de revisão por pares em revistas internacionais:

1. Influência dos parâmetros de treinamento na estabilidade dos WBFS e proposta de melhoria dos algoritmos existentes (resultados obtidos no Capítulo 5);
2. Classificação de vinhos Tempranillo e Tempranillo branco da Espanha;
3. Classificação das principais categorias comerciais de vinhos da América do Sul;
4. Classificação de solos de Oregon e Nevada.

Ademais, os seguintes estudos estão em processo de finalização da escrita para então serem submetidos:

1. Revisão e caracterização do uso de WBFS (resultados obtidos no Capítulo 3);
2. Classificação de amostras de saliva de pacientes com câncer bucal e pacientes do grupo de controle;
3. Classificação de amostras de urina de pacientes com câncer de pulmão e pacientes do grupo de controle.

1.2 Organização da tese

Esta tese está organizada em mais cinco capítulos. O Capítulo 2 abrange os conceitos sobre RNAs, métodos de seleção de variáveis baseados em RNAs e seleção de variáveis utilizados nesta tese. O Capítulo 3 apresenta uma revisão sobre o uso e caracterização dos WBFS em diferentes áreas de pesquisa. O Capítulo 4 apresenta a proposta da abordagem de votação para avaliar a estabilidade dos WBFS e os resultados obtidos. O Capítulo 5 apresenta a proposta de melhoria dos WBFS mais estáveis identificados no Capítulo 4, além de apresentar os resultados obtidos ao utilizar diferentes RNAs e parâmetros de treinamento para calcular a importância das variáveis por meio de um WBFS. Por fim, nossas conclusões e sugestões de trabalhos futuros seguem no Capítulo 6.

CAPÍTULO 2

Interpretação de Redes Neurais Artificiais

Este capítulo tem como objetivo apresentar ao leitor os principais conceitos relacionados às redes neurais artificiais e sua interpretabilidade. Também descrevemos as redes do tipo *Multilayer Perceptron* (MLP) e *Extreme Learning Machines* (ELM), e os conceitos sobre seleção de variáveis que serão utilizadas nos testes realizados ao longo desta tese.

2.1 Visão Geral

O termo *Rede Neural Artificial* (RNA) engloba uma grande família de técnicas desenvolvidas na área de aprendizado de máquina. As RNAs foram inspiradas na maneira como o cérebro humano armazena e processa as informações, representando um dos modelos mais famosos de inteligência artificial [53].

Considere um conjunto de treinamento $\mathcal{D} = \{(\mathbf{x}_i, y_i)_{i=1}^{n} | \mathbf{x}_i \in \mathbb{R}^g\}$. As n amostras do conjunto de dados são representadas por $\mathbf{x}_i$, cada uma descrita por g variáveis. A classe de dados é representada por y_i que assume valores contínuos em problemas de regressão e valores categóricos em problemas de classificação.

A RNA que utiliza o conjunto de treinamento $\mathcal{D}$ é considerada como um sistema de processamento de dados na forma de um grafo direcionado. O processamento neste sistema ocorre por meio de um conjunto de neurônios dispostos em uma ou mais camadas [53]. Os sinais são passados entre os neurônios através de ligações (sinapse) e cada neurônio funciona conforme o modelo do *perceptron*.

O *perceptron*, considerado o antecessor e a base das modernas RNAs, consiste de um único neurônio artificial que computa uma função de superfície de decisão binária, ou seja, ele é um classificador linear em que um hiperplano no espaço de entrada pode separar adequadamente as duas classes de dados. O modelo matemático do perceptron é apresentado a seguir:

$$\mathbf{o} = \sum_{i=1}^{n} f(\mathbf{x}_i^{\top} \mathbf{w}_i + b), \tag{2-1}$$

em que x_i são os dados de entrada, e estes são associados a um vetor de pesos w por meio do produto interno, e somados a uma constante b. O resultado deste cálculo é então aplicado à uma função de ativação $f(\cdot)$ que realiza uma transformação no valor computado pelo neurônio para uma determinada faixa de valores. A saída do neurônio é definida por o.

Alguns exemplos de funções de ativação incluem:

- **Função linear:** representa uma reta no plano cartesiano e não altera o valor de saída do neurônio;
- **Função degrau:** retorna o valor 0 quando o valor computado é negativo, ou retorna 1 se o valor computado for positivo, de forma análoga à um degrau no plano cartesiano;
- **Função sigmoide (ou logística):** assume o formato de S e varia de 0 a 1;
- **Função tangente hiperbólica:** similar a função sigmoide, essa função também tem um formato de S, mas varia de -1 a 1;
- **Função *softmax*:** retorna a probabilidade dos dados pertencerem a uma classe de dados.

Os pesos de conexão do perceptron são atualizados por várias iterações com base em uma taxa de aprendizagem até que a saída do neurônio seja coerente com a saída real dos dados de treinamento.

Uma rede neural é organizada em três conjuntos de camadas: a camada de entrada de dados, a(s) camada(s) oculta(s) que são constituídas cada uma de um conjunto de neurônios, e a camada de saída que determina a classe dos dados em um problema de classificação, ou um número real em um problema de regressão.

A arquitetura da rede é definida conforme os neurônios e as camadas estão dispostos. A conexão das camadas pode ocorrer em apenas uma direção, da camada de entrada para a de saída (*feedforward*), com uma ou múltiplas camadas ocultas e é considerada um aproximador universal para qualquer tipo de problema. Outra arquitetura é a rede neural recorrente, em que ocorre uma retroalimentação, no qual pode haver conexões entre quaisquer nós.

O treinamento da rede neural é definido como o método em que os pesos de conexão são ajustados. A RNA recebe um conjunto de dados e inicia um processo de treinamento para ajustar os pesos das conexões entre os neurônios de maneira a minimizar o erro da RNA.

O algoritmo mais comum para treinar as redes é o *backpropagation*, que funciona da seguinte maneira: o vetor de entrada é aplicado à RNA, a saída é calculada e em seguida comparada com a saída esperada. O erro encontrado é retroalimentado através da rede e os pesos são atualizados de acordo com um determinado algoritmo a fim de minimizar a soma dos erros quadrados.

Dentre os possíveis algoritmos para ajustar os pesos da rede neural o *backpropagation* com o método de descida de gradiente é um dos mais utilizados. Outros algoritmos como algoritmos genéticos, retropropagação resiliente (RPROP) e o algoritmo de *Levenberg Marquardt* (LM) também podem ser utilizados para ajustar os pesos na rede neural.

2.1.1 *Multilayer Perceptron*

O algoritmo *Multilayer Perceptron* (MLP) é o tipo mais comum de redes *feed-forward* [117]. Este algoritmo pode ser utilizado com uma ou mais camadas ocultas. Nesta tese utilizaremos o MLP com uma camada oculta. A primeira camada é composta pelos dados de entrada ($\mathbf{x}_i \in \mathbb{R}^g$), a camada oculta é composta por neurônios artificias, e por fim, a camada de saída é representada pelos neurônios de saída que mapeiam a classe de dados y_i.

A Figura 2.1 apresenta uma representação gráfica da arquitetura de uma rede do tipo MLP para amostras $\mathbf{x}_i \in \mathbb{R}^g$, N neurônios na camada oculta e um neurônio na camada de saída.

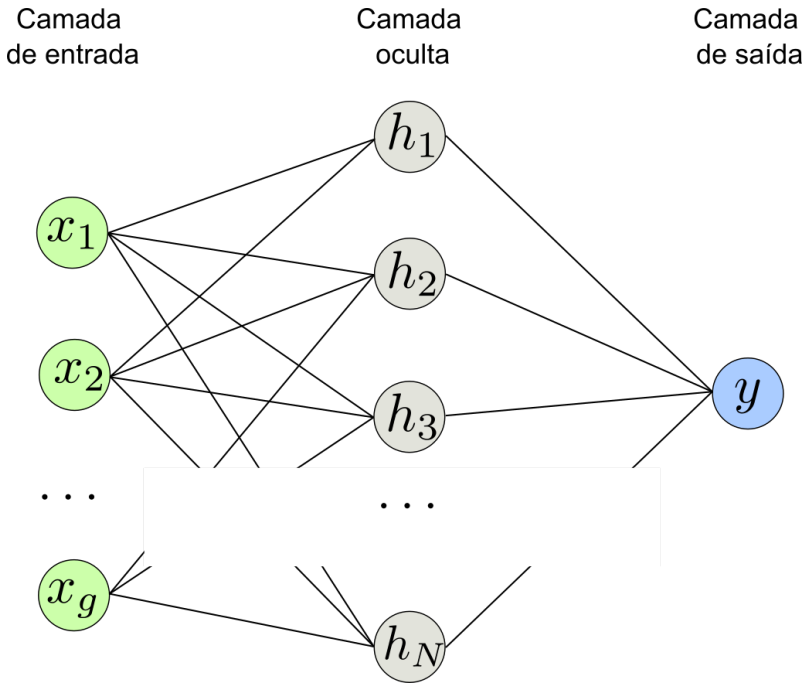


Figura 2.1: Representação gráfica de uma rede do tipo MLP com uma camada oculta.

Os neurônios ocultos e os neurônios da camada de saída realizam os cálculos necessários por meio da Equação (2-1), que descreve o funcionamento do *perceptron*. No caso do neurônio da camada de saída, o peso de conexão entre a camada oculta e a camada de saída será denominado pelo vetor $\mathbf{v}$, e a entrada dos neurônios da camada de saída será o resultado obtido nos neurônios da camada oculta.

Neste estudo utilizamos a rede do tipo MLP com *backpropagation* juntamente com a função de ativação sigmoide, representada pela equação a seguir:

$$\sigma(x) = \frac{1}{1+e^{-(x)}}. \tag{2-2}$$

O objeto do MLP é determinar uma matriz de pesos com o menor erro $E(W)$, definido conforme:

$$E(W) = \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{o}_i - \mathbf{y}_i\|^2, \tag{2-3}$$

em que $\mathbf{o}_i$ é a saída obtida pela rede neural e $\mathbf{y}_i$ é a saída esperada da i-ésima amostra do conjunto de treinamento $\mathcal{D}$.

O uso de uma técnica de regularização pode melhorar a generalização de um modelo de RNA. Nesta tese foi utilizada a técnica de regularização de *weight decay*, um dos tipos de regularização mais utilizados [53]. Esta técnica visa penalizar grandes valores dos pesos de conexão tornando-os próximo de zero. A função de erro do modelo $E(W)$ (Equação 2-3) é então modificada pela adição de um termo de penalidade para penalizar grandes valores dos pesos de conexão:

$$R(W) = E(W) + \lambda E_m(W), \tag{2-4}$$

em que o termo $E(W)$ é a função de erro definida na Equação (2-3), λ representa o parâmetro de regularização que determina a força da penalidade, conhecido como *weight decay*, e $E_m(W)$ é a penalidade da complexidade, definido por:

$$E_m(W) = \sum_{i \in \mathcal{L}_{total}} w_i^2, \tag{2-5}$$

em que $\mathcal{L}_{total}$ representa o quantidade total de pesos de conexão da RNA. Para nossos experimentos nós utilizamos os seguintes valores de *weight decay* $\lambda = (0, 0.0001, 0.001, 0.01, 0.1, 0.3, 0.5)$, valores esses comumente utilizados na literatura. Quando $\lambda = 0$ o treinamento dos dados ocorre normalmente sem a regularização. Por outro lado, quando λ assume valores grandes o processo de treinamento das amostras se torna não confiável [53].

2.1.2 *Extreme Learning Machines*

O algoritmo *Extreme Learning Machines* (ELM) foi proposto por Huang e colaboradores [59] com o objetivo de reduzir o tempo computacional gasto no processo de *backprogation*. O ELM é baseado no princípio de que os pesos $\mathbf{w}$ e a constante b são determinadas de forma aleatória e os pesos $\mathbf{v}$ que conectam a camada oculta com a camada

de saída são calculados em um único passo. Essa abordagem prevê uma rede neural com boa capacidade de generalização e extremamente rápida.

Considere um conjunto de n amostras arbitrárias $(\mathbf{x}_j, \mathbf{y}_j) \in \mathbb{R}^g \times \mathbb{R}^u$, ou seja, n amostras descritas em g variáveis associadas a uma das u classes que existem no banco de dados. A rede neural ELM com h neurônios na camada oculta pode ser representada pela Equação (2-6) a seguir:

$$\sum_{i=1}^{h} \alpha_i f(\mathbf{x}_j) = \sum_{i=1}^{h} \alpha_i f(\langle \mathbf{w}_i, \mathbf{x}_j \rangle + b_i) = \mathbf{o}_j, \quad j = 1, \cdots, n \tag{2-6}$$

em que $\mathbf{w}$ representa os pesos do i-ésimo neurônio oculto associados aos dados de entrada, b representa uma constante, $f(\cdot)$ denota a função de ativação da rede, α representa o peso da conexão do i-ésimo neurônio a camada de saída e $\mathbf{o}$ representa a saída da rede neural.

Partindo da possibilidade de que a Equação (2-6) possa aproximar o valor $\mathbf{o}_i$ da classe real $\mathbf{y}_i$ das n amostras, então existe um $(\mathbf{w}_i, b_i)$ e um α_i que satisfazem a Equação (2-7) [58]:

$$\sum_{j=1}^{h} \|\mathbf{y}_j - \mathbf{o}_j\| = 0. \tag{2-7}$$

Essa possibilidade pode ser representada por:

$$\mathbf{H}\alpha = \mathbf{O}, \tag{2-8}$$

em que $\mathbf{H}$, a matriz de saída da camada oculta da rede neural, é multiplicada pelos pesos α de conexão entre a camada oculta e camada de saída, resultando na matriz de saída $\mathbf{O}$. Cada elemento da Equação (2-8) é definido por:

$$\mathbf{H} = \begin{bmatrix} f(\langle \mathbf{w}_1, \mathbf{x}_1 \rangle + b_1) & \cdots & f(\langle \mathbf{w}_h, \mathbf{x}_1 \rangle + b_h) \\ \vdots & \ddots & \vdots \\ f(\langle \mathbf{w}_1, \mathbf{x}_n \rangle + b_1) & \cdots & f(\langle \mathbf{w}_h, \mathbf{x}_n \rangle + b_h) \end{bmatrix}_{n \times h}, \alpha = \begin{bmatrix} \alpha_1^T \\ \vdots \\ \alpha_h^T \end{bmatrix}_{h \times u}, \mathbf{O} = \begin{bmatrix} \mathbf{o}_1^T \\ \vdots \\ \mathbf{o}_n^T \end{bmatrix}_{n \times u}. \tag{2-9}$$

A i-ésima coluna de $\mathbf{H}$ representa o i-ésimo neurônio da camada oculta com os respectivos valores de entrada. Se o número de neurônios h da camada oculta for igual ao número de amostras n, a matriz $\mathbf{H}$ é quadrada e a Equação (2-8) pode ser satisfeita com um erro zero [59]. Entretanto, não há garantia de que o valor de h seja igual a n.

Nesses casos, a solução do sistema é dada pelo uso da matriz pseudo-inversa Moore-Penrose ($\mathbf{H}^+$), permitindo um número $h < n$, conforme a Equação (2-10):

$$\alpha = \mathbf{H}^+ \mathbf{O} \tag{2-10}$$

A pseudo-inversa da matriz $\mathbf{H} \in \mathbb{R}^{n \times h}$ é a matriz $\mathbf{H}^{+} \in \mathbb{R}^{h \times n}$, uma matriz única [58]. Os passos para a execução de uma rede neural ELM se resumem a:

1. Atribuir aleatoriamente os pesos $\mathbf{w}_i$ e b_i;
2. Calcular a matriz $\mathbf{H}$;
3. Calcular os pesos α, em que $\alpha = \mathbf{H}^{+}\mathbf{O}$.

Dessa maneira, a rede neural ELM é capaz de realizar um aprendizado rápido e com uma boa capacidade de generalização ao aproximar os valores **o** de **y**.

As funções de ativação utilizadas em conjunto com esse modelo de rede neural foram as funções linear (Equação 2-11), tangente sigmoide (Equação 2-12), função de base radial (Equação 2-13) e função seno (Equação 2-14).

$$f(x) = x \tag{2-11}$$

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2-12}$$

$$f(x) = exp(-\gamma \|x - x_i\|^2) \tag{2-13}$$

$$f(x) = seno(x) \tag{2-14}$$

2.1.3 Vantagens e desvantagens

As RNAs possuem três importantes características consideradas vantagens [132]: a primeira diz respeito à sua capacidade de adquirir informações durante a fase de treinamento dos dados; a segunda, refere-se à forma compacta de armazenamento do conhecimento adquirido, representado por uma matriz de pesos e constantes; e por último, a sua capacidade de encontrar a solução de um determinado problema na presença de um conjunto de dados com ruídos.

No entanto, as RNAs são difíceis de interpretar, e consequentemente, identificar quais variáveis são as mais importantes também é uma tarefa difícil devido ao aprendizado da RNA ser representado por uma matriz de pesos. Além disso, é impraticável apontar como as variáveis se relacionam com o modelo criado, o que por outro lado, pode ser facilmente realizado em modelos lineares [163]. Essa é uma das características das RNAs que por vezes inviabiliza seu uso em aplicações que precisam de resultados interpretáveis.

Considere o exemplo de dados médicos. Em uma aplicação que objetiva predizer se um paciente terá uma possível recaída para a depressão, o médico precisa saber quais os fatores que indicam a recaída em um estágio inicial para poder tratar o paciente. Saber

apenas se o paciente terá uma recaída ou não, sem saber quais os fatores que indicam tal cenário inviabiliza um tratamento de prevenção. Esse é um exemplo de aplicação que apenas um classificador com uma alta capacidade de predição não é o suficiente, sendo necessário meios de descobrir padrões úteis que levam ao resultado da predição.

De maneira similar, em uma aplicação de mineração de dados para predizer o sucesso ou falência de uma empresa, os administradores precisam saber quais os aspectos e características que mais contribuem para o sucesso ou fracasso, de maneira a trabalhar nesses aspectos e características.

Além disso, para um determinado conjunto de dados e uma determinada arquitetura de rede neural podem haver múltiplos modelos com diferentes pesos e funções de ativação que resultam em predições similares para o mesmo problema. Esse cenário dificulta um pouco mais o entendimento da rede neural e das variáveis relevantes para o modelo. Isso porque, para cada modelo, os valores dos pesos de conexão podem ser diferentes, mesmo que tenham sido obtidos por uma mesma arquitetura nos mesmos dados de treinamento [163].

2.2 Interpretação

Um dos aspectos reportados em diversos artigos sobre a dificuldade de se utilizar as RNAs é a sua complexidade e inviabilidade de explicar como ocorrem as relações entre as variáveis e a saída dos dados [163, 92, 101, 165, 43, 86]. Isso ocorre devido a rede neural ser um modelo não linear capaz de modelar problemas complexos, porém, a interpretação do resultado obtido é uma tarefa complexa.

Ademais, grande parte das análises de dados reais requerem resultados explicativos ao invés de apenas métricas que representam o desempenho do modelo. Uma RNA é considerada um modelo do tipo “caixa preta” ou do tipo “não-transparente”, o que significa que a partir de uma entrada obtemos uma saída, porém o que acontece entre a entrada e a saída não é um processo transparente [38].

Esse problema da interpretação das RNAs não é causado por ausência de informação, mas sim pela abundância de informações representadas nas conexões da rede que são difíceis de serem interpretadas [84]. Ou seja, não há uma maneira de entender as decisões tomadas pela rede neural observando apenas o peso das conexões. Este aspecto é considerado uma das desvantagens de se utilizar as RNAs [84]. No entanto, este aspecto não torna a rede neural menos capaz de modelar problemas e obter bons resultados.

Neste contexto, surgiram alguns métodos para estimar a importância das variáveis de entrada em um modelo de RNA em relação aos neurônios de saída da rede, descritos a seguir.

2.3 Abordagens para extrair conhecimento de RNAs

Existem três abordagens de algoritmos para a interpretação das redes neurais: determinação de um valor de importância das variávei, análise de sensitividade e extração de regras.

Os modelos que determinam um valor de importância das variáveis realizam essa tarefa com base nos pesos de conexão entre os neurônios da rede para calcular um valor que indique a relação entre as entradas e as saída da rede neural. Chamaremos esses algoritmos de *weight based feature selection - WBFS*. Esses algoritmos serão descritos na próxima seção.

A análise de sensitividade diz respeito aos efeitos da variação nas entradas do modelo na saída do modelo. Estes algoritmos funcionam variando cada variável de entrada sobre o intervalo total ou parcial de valores possíveis, mantendo todas as outras variáveis de entrada constantes em um percentil especificado, para que a importância das variáveis de interesse possa ser avaliado pela variação induzida na saída da rede [149].

Por fim, a extração de regras visa extrair conhecimento de uma rede neural a partir de regras de decisão. Apesar de regras do tipo “se - então - senão” serem facilmente interpretáveis, elas não ilustram a importância relativa de cada variável, indicando apenas um caminho para a classificação. Ademais, as regras de decisão não demonstram o quão sensível a saída da rede é em relação à uma determinada variável, além de serem rígidas e inflexíveis, e de existir a possibilidade de gerar um número muito grande de regras em um complexo banco de dados, podendo dificultar ainda mais a interpretação das variáveis [165].

Até a data de publicação dos últimos estudos que avaliaram os métodos de seleção de variáveis baseado em redes neurais, não foi estabelecido um consenso sobre qual grupo de métodos (regras de decisão, valor de importância e análise de sensibilidade) supera o outro [43, 92].

Os métodos baseados nos pesos de conexão podem ser considerados como seletores de variáveis do tipo filtro, pois resultam em um ranking de importância. A seguir os métodos existentes dessa categoria são descritos.

2.3.1 Métodos baseados nos pesos de conexão

Considere um conjunto de treinamento $\mathcal{D} = \{(\mathbf{x}_i, y_i)_{i=1}^{n} | \mathbf{x}_i \in \mathbb{R}^g\}$ com n amostras, cada uma descrita por g variáveis. Após o treinamento da rede neural no conjunto de dados $\mathcal{D}$, um WBFS pode ser aplicado nas matrizes de pesos resultantes, sendo uma matriz composta dos pesos de conexão ($\mathbf{w}$) entre a camada de entrada e a camada oculta, e uma matriz $\mathbf{v}$, que representa os pesos de conexão entre a camada oculta e a camada de saída da rede. A Figura 2.2 apresenta a arquitetura de uma rede neural com 3 neurônios

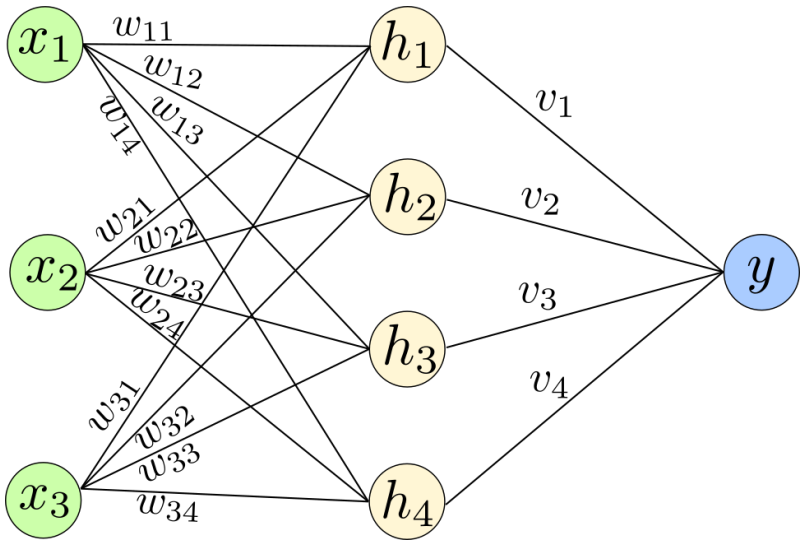


Figura 2.2: Representação gráfica dos pesos de conexão de uma rede neural com arquitetura 3-4-1.

de entrada, 4 neurônios na camada oculta e 1 neurônio de saída (arquitetura 3-4-1), com a identificação de seus respectivos pesos de conexão.

O primeiro algoritmo proposto para estimar a importância das variáveis com base nos pesos da rede neural foi proposto por Garson em 1991 [42]. Este algoritmo utiliza dos pesos **w** e **v** conforme a seguir:

$$Garson_{ik} = \frac{\sum_{j=1}^{h} \frac{|w_{ij}||v_{jk}|}{\sum_{i=1}^{g}|w_{ij}|}}{\sum_{i=1}^{g}\sum_{j=1}^{h} \frac{|w_{ij}||v_{jk}|}{\sum_{i=1}^{g}|w_{ij}|}}, \tag{2-15}$$

em que w_{ij} representa o peso de conexão entre a i-ésima variável e o j-ésimo neurônio da camada oculta, h representa o número de neurônios na camada oculta, g representa o número de variáveis de entrada, e v_{jk} representa o peso de conexão entre o j-ésimo neurônio da camada oculta e o k-ésimo neurônio da camada de saída.

Este algoritmo não considera a direção da contribuição das variáveis para a saída da rede (positiva/negativa) devido ao uso de módulos, nem o valor do *bias*, comumente representado por w_{0j} para a camada oculta e v_{0j} na camada de saída.

O segundo algoritmo foi proposto por Yoon em 1994 [154]. Este algoritmo é descrito pela equação a seguir, em que os símbolos possuem o mesmo significado apresentado para o algoritmo de Garson.

$$Yoon_{ik} = \frac{\sum_{j=1}^{h} w_{ij}v_{jk}}{\sum_{i=1}^{g}\left|\sum_{j=1}^{h} w_{ij}v_{jk}\right|} \tag{2-16}$$

Este algoritmo leva em consideração a direção da contribuição da variável para

a saída da rede, além de ser mais simples do que o algoritmo de Garson [165].

O terceiro algoritmo foi proposto por Howes e Crook em 1999 [57], definido por:

$$Howes_{ik} = \frac{\sum_{j=1}^{h} \left| \left(\frac{w_{ij}}{\sum_{i=0}^{g} |w_{ij}|} \right) * v_{jk} \right|}{\sum_{j=0}^{h} v_{jk}} \tag{2-17}$$

em que o índice 0 em w_{0j} e v_0 indica o *bias* do j-ésimo neurônio da camada oculta e o bias do k-ésimo neurônio da camada de saída. Observamos que o *bias* é considerado na normalização, mas não nos efeitos das variáveis de entrada.

O quarto algoritmo foi proposto por Tsaur em 2002 [137], definido pela Equação (2-18).

$$Tsaur_{ik} = \frac{\frac{\sum_{j=1}^{h}(w_{ij} + v_{jk})}{h}}{\sum_{i=1}^{g} \frac{\sum_{j=1}^{h}(w_{ij} + v_{jk})}{h}} \tag{2-18}$$

Esse método foi comparado com os métodos de Garson e Yoon no estudo de Wong et al. [146]. Os resultados apresentados demonstram que este algoritmo obteve uma ordem de importância diferente da ordem similar encontrada com Garson e Yoon, em que esses dois últimos resultaram em um melhor desempenho.

Por fim, o quinto algoritmo foi proposto por Olden em 2004 [92]. Este algoritmo é comunmente chamado de *connection weights method*, sendo definido por

$$Olden_{ik} = \sum_{j=1}^{h} w_{ij} v_{jk} \tag{2-19}$$

O algoritmo de Olden possui o mesmo numerador do algoritmo de Yoon definido na Equação (2-16). Dessa forma, o algoritmo de Yoon pode ser considerado uma normalização do algoritmo de Olden.

É importante ressaltar que cada uma das equações que definem os algoritmos de Garson, Olden, Yoon, Tsaur e de Howes e Crook resultam em um valor de importância de cada variável para cada neurônio de saída. Este aspecto não é abordado em nenhum dos artigos que propuseram esses algoritmos. As equações originais não levam em consideração a possibilidade de múltiplos neurônios na camada de saída.

Ao perceber esse aspecto, decidimos por calcular a importância final de cada variável com base na média das importâncias obtidas nos neurônios de saída para cada variável. Mais detalhes sobre este aspecto será discutido no próximo capítulo que apresenta uma revisão sobre o uso dos WBFS.

2.4 Seleção de variáveis

O princípio de identificar a importância das variáveis de entrada em uma rede neural para torná-la mais interpretável é similar à seleção de variáveis, etapa de pré-processamento comumente utilizada na mineração de dados. Os WBFS tem sido utilizados com frequência em aplicações de diversas áreas do conhecimento, em que, primeiro, os autores utilizam uma RNA para realizar uma classificação ou regressão e, em seguida, avaliam a importância das variáveis com base na rede neural construída (os dados que sustentam essa afirmação são apresentados no próximo capítulo). Esta etapa de identificar a importância das variáveis pode ser realizada por meio de diferentes algoritmos de seleção de variáveis.

Inevitavelmente os conjuntos de dados reais possuem variáveis redundantes e irrelevantes, assim como possuem variáveis relevantes que fazem sentido para as classes de dados. Entretanto, identificar manualmente quais são as variáveis relevantes pode ser uma tarefa exaustiva, visto que em um conjunto de dados com g variáveis existem 2^g possíveis subconjuntos de variáveis, caracterizando um problema NP-difícil [16].

O conhecimento prévio sobre o domínio do problema pode auxiliar a selecionar manualmente algumas das variáveis que possivelmente já foram reportadas como úteis para a solução daquele problema específico. Contudo, nem sempre essa é uma solução viável devido a particularidade de cada conjunto de dados e devido à possibilidade de o conjunto de dados possuir uma alta dimensionalidade, o que torna o número de possíveis combinações exponencial, inviabilizando o teste de todos os subconjuntos de variáveis. Dessa maneira, cabe aos métodos de seleção de variáveis eliminar as variáveis que não auxiliam a solucionar o problema.

A seleção de variáveis é uma das etapas de pré-processamento da mineração de dados, que visa selecionar um grupo de variáveis do conjunto de dados baseado em algum critério, selecionando assim variáveis relevantes para o conjunto de dados [14]. Em alguns conjuntos de dados, as variáveis que não têm correlação com as classes servem apenas como ruído e podem introduzir viés no modelo de classificação e reduzir seu desempenho [16].

Podemos citar as seguintes vantagens ao usar apenas um subconjunto de variáveis ao invés de utilizar todo o conjunto:

- Compreensão do conjunto de dados visto que as variáveis são relevantes para a discriminação das amostras;
- Redução do tempo computacional necessário para processar os dados; e,
- Aumento da capacidade de predição.

A seleção de variáveis é constantemente abordada na mineração de dados como uma maneira de tratar a alta dimensionalidade de dados, cenário em que existe uma grande

quantidade de variáveis que caracterizam a base de dados. Utilizar um grande número de variáveis em uma análise de dados requer um alto poder computacional, além de aumentar as chances de um modelo produzir falsos padrões, devido à utilização de variáveis que não auxiliam a encontrar a solução do problema.

Diante de um grande conjunto de dados é comum a aplicação dos métodos de extração de variáveis em vez dos métodos de seleção. A extração de variáveis realiza uma combinação das variáveis originais para dar origem a novas variáveis, chamadas, por exemplo, de componentes na técnica de Análise de Componentes Principais (PCA). Alguns exemplos de algoritmos que realizam extração de variáveis são: PCA, Análise de Componentes Independentes (ICA), e Transformações *Walvelet*.

O uso de extração de variáveis pode resultar em uma melhor capacidade discriminativa do que o uso do melhor subconjunto de variáveis, mas as novas variáveis, que são uma combinação linear ou não-linear de todas as variáveis, podem não ter um significado claro [60]. Isso dificulta a compressão dos resultados e desfavorece a descoberta de padrões úteis.

Por outro lado, para a classificação de imagens a extração de variáveis é uma ferramenta indispensável, sendo que outras ferramentas como a seleção de variáveis podem ser aplicadas caso necessário em uma fase posterior à extração de variáveis [108, 69].

Ademais, cerca de 80% dos recursos utilizados em uma aplicação de mineração de dados é destinado à limpeza e ao pré-processamento de dados [101]. A seleção de variáveis, uma das fases de pré-processamento, está dentro da maior parte de recursos empregados em uma análise de dados, o que a torna uma fase importante, influenciando diretamente o resultado da análise.

Essa influência pode ser medida em termos de precisão do modelo de classificação gerado e em termos de informações úteis encontradas. Exemplos de influência com base na precisão do modelo de classificação ou com base nas informações úteis são: identificar qual o conjunto de propriedades físico-químicas é responsável por classificar a classe de dados de produtos alimentícios (orgânico x convencional, região, tipo de matéria prima utilizada na produção, entre outros); e identificar quais os sintomas que são predominantes para a predição de doenças ou recaídas em transtornos mentais, assim como quais os fatores que melhor caracterizam um determinado grupo alunos em um conjunto de dados educacionais.

No caso dos WBFS, as RNAs são treinadas com base em todas as variáveis do conjunto de dados, e após estabelecer um modelo de classificação ou de regressão, a importância das variáveis para tal tarefa é estimada. O resultado obtido é um ranking de importância, que também é o resultado obtido ao utilizar uma seleção de variáveis do tipo filtro, explicado na próxima seção.

2.4.1 Tipos de seleção de variáveis

A seleção de variáveis pode ocorrer por métodos filtro, *wrapper* ou embutidos. Um método filtro atribui um valor de importância (*score*) para cada variável de maneira a construir um *ranking*, no qual as variáveis de maior *score* são as variáveis mais relacionadas com a classe de dados.

Os métodos *wrapper* selecionam as variáveis com base no desempenho de um classificador em um dado subconjunto de variáveis determinados por um algoritmo de pesquisa (como *forward selection* que adiciona as variáveis gradativamente e *backward elimination*, que elimina as variáveis gradativamente). Com um classificador escolhido, vários subconjuntos de variáveis são utilizados como dados de entrada ao classificador. O subconjunto de variáveis que resultar na maior acurácia (métrica que define a capacidade de um modelo de classificação acertar uma predição) é escolhido.

Por fim, o métodos embutidos incluem a seleção de variáveis como parte do treinamento do modelo, como ocorre na Eliminação Recursiva de Variáveis (RFE) e no *Least Absolute Shrinkage and Selection Operator* (LASSO).

Conforme mencionado na seção anterior, os WBFS se comportam de maneira semelhante aos métodos de seleção de variáveis do tipo filtro, que resultam em um ranking de importância. A diferença entre os métodos do tipo filtro como por exemplo *Relief*, *F-score*, *Chi-squared* e *Random Forest Importance*, é que estes algoritmos são computados com base nos valores das amostras para cada variável enquanto os WBFS são computados com base nos pesos de conexão de uma rede neural. Apesar dessa diferença, esses algoritmos obtém o mesmo resultado, um ranking de importância.

Os rankings de importância são comumente utilizados para realizar inferências sobre os dados, e também como uma ferramenta para gerar subconjuntos de variáveis para serem avaliados posteriormente, como por exemplo, utilizar os subconjuntos de variáveis como entrada de um modelo de aprendizado de máquina.

A ação de utilizar diferentes subconjuntos de variáveis como entrada de algoritmos de aprendizado de máquina e escolher o subconjunto com a melhor capacidade de predição caracteriza um método *wrapper*, descrito no início desta seção. Após determinar um valor de importância para cada variável, essas são ordenadas do maior para o menor. A partir daí é possível escolher um subconjunto de variáveis que obteve os maiores valores de importância por meio de um ponto de corte. Este ponto de corte pode ser definido de diversas maneiras, como por exemplo:

- Escolher as i primeiras variáveis;
- Escolher uma determinada porcentagem de variáveis;
- Escolher as variáveis que apresentam uma maior diferença de valores de importância;

- Utilizar *backward elimination* ou *forward selection* para classificar cada subconjunto de dados gerado.

Obter *rankings* de importância possibilita também a criação de *i* subconjuntos de dados com as *i*-ésimas variáveis mais importantes. Cada subconjunto de dados pode ser aplicado a um classificador, permitindo assim inferir como o classificador se comporta ao adicionar variáveis de forma gradativa e identificar o melhor subconjunto de variáveis baseando-se na precisão obtida em cada uma das *i* classificações.

Essa abordagem de seleção de variáveis por métodos filtro em conjunto com classificadores foi utilizada por nós nos estudos realizados ao longo do doutorado de aplicação de mineração de dados em dados reais. Esses estudos incluem: classificação de vinhos Syrah [23], Tannat [17], Merlot [19] e Cabernet Sauvignon [24] de diferentes regiões, análise de concentração de bisfenóis, clorofenóis, parabenos e benzofenonas em urina de crianças Brasileiras para determinar o dano no DNA [113], classificação de vinhos produzidos com diferentes tipos de uvas no Chile [22], e classificação de vinhos da América do sul com base em características sensoriais [18].

2.5 Considerações finais

Os algoritmos de seleção de variáveis baseados nos pesos das RNAs, chamados nesta tese de WBFS - *weight based feature selection*, são o alvo desta tese de doutorado.

Essa categoria de algoritmos foi escolhido devido aos valores serem facilmente interpretados, e, caso exista um número grande de variáveis, as que possuem um valor importância menor podem ser desconsideradas da análise, visto que é comum considerarmos um ponto de corte para escolher as melhores variáveis em um método do tipo filtro.

O próximo capítulo apresenta um estudo de revisão bibliográfica envolvendo mineração de dados para a caracterização do uso dos WBFS.

CAPÍTULO 3

Caracterização do uso de WBFS

Este capítulo apresenta uma visão geral de como os algoritmos WBFS são utilizados por pesquisadores. Após o levantamento dos dados, alguns padrões e potenciais problemáticas ao utilizar um algoritmo WBFS para estimar a importância das variáveis foram identificados.

3.1 Motivação

O algoritmo de Garson foi o primeiro a ser proposto, em 1991, para avaliar a importância das variáveis de entrada em relação à saída da RNA com base nos pesos de conexão. A partir de então, outros algoritmos como os propostos por Olden, Yoon, Tsaur, e Howes e Crook surgiram com variações na maneira de lidar com os pesos de conexão das RNAs para estimar a importância das variáveis. Apesar de existir um grande número de publicações no decorrer de 29 anos desde a proposta do primeiro WBFS, até onde sabemos, não existe um estudo que revisa a aplicação desses métodos.

Dessa forma, este capítulo apresenta uma caracterização do uso dos WBFS a partir das publicações ocorridas no período de 2015 até outubro de 2020. Nós utilizamos de análise bibliométrica e de mineração de texto para obter conhecimento útil sobre os artigos encontrados em uma ampla variedade de campos de estudo. Ademais, nós fornecemos uma categorização das formas de uso dos WBFS com base na análise manual das publicações.

3.2 Metodologia

O estudo realizado neste capítulo foi conduzido seguindo os seguintes passos:

Passo 1. Seleção dos artigos;
Passo 2. Leitura e extração de dados;
Passo 3. Download dos metadados dos artigos;
Passo 4. Análise bibliométrica;

Passo 5. Mineração de texto;

Passo 6. Avaliação dos resultados.

A Subseção 3.2.1 apresenta as ações e critérios escolhidos para determinar a fonte e coleta de dados, que representam os passos 1 - 3. Em seguida, a Subseção 3.2.2 descreve as ferramentas utilizadas para a análise bibliométrica e para a mineração de texto.

3.2.1 Fonte e coleta de dados

Os artigos científicos selecionados foram publicados no período de 2015 até outubro de 2020. O objetivo da análise destes artigos foi identificar como os WBFS são utilizados na literatura. A descrição detalhada dos resultados obtidos pelos artigos não é abordada neste capítulo, visto que essa descrição foge do escopo deste estudo, trazendo diversas informações não relacionadas a esta tese de doutorado, sendo essas muito específicas para cada área de estudo que utilizou os WBFS nos últimos anos.

Poucos artigos foram encontrados a partir de uma *string* de busca combinando os termos "*neural networks* AND *feature selection*" e um dos termos específicos *"Garson algorithm"*, *"Olden algorithm"*, *"connection weights"*,*"general influence"*, *"Yoon algorithm"*, *"Tsaur"*, e *"Howes and Crook"*.

No entanto, o estudo de Garson possui mais de 1520 citações de acordo com o *Google Scholar* (ao final de outubro de 2020), o estudo de Olden possui mais de 670 citações, Tsaur possui 208 citações, o estudo de Yoon possui 160, e por fim, o estudo de Howes e Crook possui 58 citações. Mediante o fato de que existem publicações que se basearam nos artigos que propuseram os algoritmos aqui estudados, levantamos a hipótese de que os artigos de pesquisa que utilizaram esses WBFS não indexaram os artigos com os termos pesquisados, pois os métodos não eram o foco dos artigos, mas sim a aplicação.

Dessa maneira, o estudo aqui apresentado foi baseado nos artigos que citaram os artigos originais de Garson, Olden, Yoon, Tsaur e de Howes e Crook. A busca foi realizada a partir dos resultados de artigos que citaram o artigos originais pelo *Google Scholar* durante o período 2015 - 2020. Neste período foram publicados 761 artigos que citaram os artigo de Garson, 419 que citaram o artigo de Olden, 88 que citaram o artigo de Yoon, 30 que citaram o artigo de Tsaur, e 10 artigos que citaram o artigo de Howes e Crook.

Após a identificação dos artigos de interesse, nós realizamos uma busca manual por cada artigo para encontrar os trabalhos que aplicaram os WBFS. Consideramos apenas os artigos que relatam a aplicação dos WBFS para a resolução de um problema. Além deste critério, utilizamos os seguintes critérios de exclusão:

- Acesso indisponível;

- Artigos de revisão;
- Livros e capítulo de livros;
- Citação do artigo buscado em outro contexto;
- Teses e dissertações;
- Artigos que não foram escritos em inglês; e,
- Duplicações.

Após analisar cada artigo e excluir alguns conforme a lista de critérios acima, foram armazenadas 356 publicações. Vale destacar que, durante a análise dos artigos que citaram o algoritmo de Garson, encontramos estudos que também utilizaram o algoritmo de Olden. No entanto, alguns destes estudos não constavam na busca pelos artigos que citaram o algoritmo de Olden. Isso ocorreu devido ao fato que alguns estudos citaram o artigo de Olden que foi publicado em 2002 [91] em vez do estudo que foi publicado em 2004 [92]. A proposta do método de Olden ocorreu apenas na publicação de 2004. Na publicação de 2002 os autores compararam o algoritmo de Garson, o diagrama de interpretação de rede neural e análise de sensitividade. Dessa forma, as publicações que citaram o estudo de Olden de 2002 e que também citaram o estudo de Garson ou o estudo de Olden de 2004 foram incluídos. No entanto, as publicações que citaram apenas o estudo de Olden de 2002 não foram incluídos neste estudo.

A leitura dos artigos selecionados possibilitaram a criação de uma base de dados contendo as seguintes informações:

- Título;
- Autor;
- Ano de publicação;
- Tipo de RNA utilizada;
- Função de ativação utilizada na camada oculta e na camada de saída;
- Algoritmo de treinamento;
- Quantidade de variáveis de entrada;
- Algoritmo WBFS utilizado;
- Número de camadas ocultas;
- Quantidade de RNAs que foram modeladas; e,
- Quantidade de RNAs utilizadas para calcular o WBFS.

Após a construção desta base de dados, buscamos os metadados das publicações na base *Web of Science* (WoS). Quarenta e cinco dos 356 artigos não constavam na base WoS e foram excluídos da análise de dados, restando assim, 311 publicações.

3.2.2 Análise de dados

A análise de dados aqui apresentada foi realizada com base em princípios de diferentes áreas. A área principal que guiou essa pesquisa foi a mineração de dados, seguida por análise bibliométrica e mineração de texto.

A mineração de dados é uma das etapas do processo de descoberta de conhecimento em bases de dados (*Knowledge Discovery in Database* - KDD) [34]. Este processo é caracterizado pelas fases de seleção dos dados, pré-processamento, transformação, mineração de dados e avaliação dos resultados. Ao minerar dados, o pesquisador pode retornar e avançar entre essas fases até que consiga identificar padrões úteis. Motivados por essa rotina de análise de dados e possíveis padrões úteis que podem ser encontrados, organizamos os dados e planejamos os experimentos.

Neste contexto, nós selecionamos os artigos, armazenamos os metadados e selecionamos manualmente as informações potencialmente relevantes para a caracterização do uso dos WBFS. Em seguida, os metadados dos artigos extraídas do WoS foram organizados em um arquivo com extensão .bib. Toda a análise foi realizada com o software R, e, após a leitura dos dados .bib, as informações que foram extraídas manualmente dos artigos foram concatenadas ao *data frame* contendo os metadados dos artigos. Em seguida, a extração de conhecimento e padrões foi realizada por meio de técnicas de análise bibliométrica e mineração de texto.

Análise bibliométrica

A análise bibliométrica é utilizada para obter informações quantitativas sobre publicações ao longo do tempo [31]. As informações obtidas são relacionadas à quantidade de publicações por ano, aspectos institucionais ou geográficos, indicadores de desempenho, disciplinas e áreas correlatas, e informações relacionadas a autorias, veículos de publicação e citações.

As informações obtidas pela análise bibliométrica podem ser comparadas à tarefa de descrição, possível de ser realizada pela mineração de dados. Segundo Fayyad et al. [34], as duas tarefas principais da mineração de dados são a predição e a descrição. A tarefa de predizer está relacionada a prever um valor numérico (regressão) ou um valor qualitativo (classificação) com base em dados de treinamento. Por outro lado, a descrição objetiva encontrar padrões que descrevem os dados que sejam interpretáveis.

A descrição de um conjunto de dados pode acorrer por meio de diferentes técnicas, dentre elas algoritmos de aprendizado de máquina supervisionados ou não supervisionados, além do uso de técnicas como seleção ou extração de variáveis. No entanto, é possível realizar uma descrição dos dados sem utilizar de técnicas sofisticadas de aprendizado de máquina. Um exemplo é a sumarização, que visa estabelecer uma

descrição compacta dos dados. Essa descrição pode ser realizada por métodos estatísticos básicos, como medidas de tendência central, medidas de dispersão e análise da frequência de valores [34].

Dessa forma, utilizamos o pacote *bibliometrix* para realizar a descrição dos artigos selecionados, visando à quantificação de publicações por ano, por tipo de algoritmo utilizado e a quantidade de publicações por área de atuação. Alguns aspectos comumente reportados em pesquisas de análise bibliométrica como a colaboração entre países, principais autores, *h-index*, rede de citações e veículos de publicações não foram considerados neste estudo, visto que essas informações não trazem conhecimento útil para a caracterização do uso dos métodos WBFS.

Mineração de texto

A mineração de texto tem como objetivo a análise computacional de dados textuais. A mineração de texto requer menos tempo e trabalho humano para analisar um texto, tornando-a mais vantajosa do que a análise manual de textos [50]. Os passos da mineração de texto incluem a organização dos dados, pré-processamento, extração de variáveis e a análise dos resultados. A mineração de texto tem sido utilizadas para analisar o perfil de vagas de emprego [95], analisar informações armazenadas em redes sociais [126, 114], avaliar o impacto do humor expressado em redes sociais no preço de ações [118], analisar o histórico e tendências no uso de irradiância solar e previsão de energia fotovoltaica [152], entre outros.

Os campos de título, resumo e palavras-chave foram utilizados nesta etapa do estudo baseado em mineração de texto. Utilizamos as seguintes tarefas de pré-processamento: transformação de letras maiúsculas para minúsculas, remoção de espaços em branco, remoção de pontuação, remoção de *stopwords* (palavras comuns que não oferecem conhecimento, como por exemplo *"and"*, *"by"*, *"its"*, *"how"*, *"what"*, entre outros) e a redução de prefixos e sufixos. Após transformar os dados, utilizamos análise de grupos (*clustering*) com base na distância entre as frequências de palavras.

Os dados dos documentos foram organizados no chamado "*Bag of Words*" contendo todos os documentos. Esses documentos foram organizados em uma matriz de termos de documento (DTM) para cada análise de texto (título, resumo, palavras-chave), em que cada linha representa um documento e cada coluna representa uma palavra ou frase [73]. Cada célula contém a frequência de ocorrência das palavras nos documentos.

Nós utilizamos a análise de *cluster* hierárquico, um algoritmo que inicialmente atribui cada amostra a seu próprio *cluster* e então o algoritmo prossegue iterativamente agrupando as amostras em novos *clusters*. Em cada iteração o algoritmo agrupa dois ou mais *clusters* semelhantes, até que haja apenas um único *cluster* composto de todas as amostras [150].

Algoritmos como os *clusters* hierárquicos permitem uma melhor visualização e compreensão dos dados devido à simplicidade do gráfico, sendo este um dendrograma que mostra a organização das amostras e suas relações em forma de árvore. Essa representação permite detectar potenciais semelhanças entre as amostras.

3.3 Resultados

3.3.1 Caracterização dos estudos selecionados

Os artigos avaliados utilizaram ao menos um dos algoritmos: Garson, Olden, Yoon, Tsaur e/ou Howes. A partir deste ponto da tese, chamaremos o algoritmo de Howes e Crook apenas por "Howes" por fins de simplicidade. Além desses algoritmos, alguns estudos também utilizaram algoritmos de análise de sensitividade, como a análise de sensitividade clássica (fixação dos valores de uma variável no valor de sua média ou algum quartil e avaliação das variações da performance da RNA), ou outros métodos como a análise de Derivadas Parciais (*PaD*), Pertubação (*Perturb*), e Perfil de Lek (*Profile*). A Figura 3.1 mostra a quantidade de estudos que utilizaram cada uma das técnicas, incluindo as análises de sensitividades.

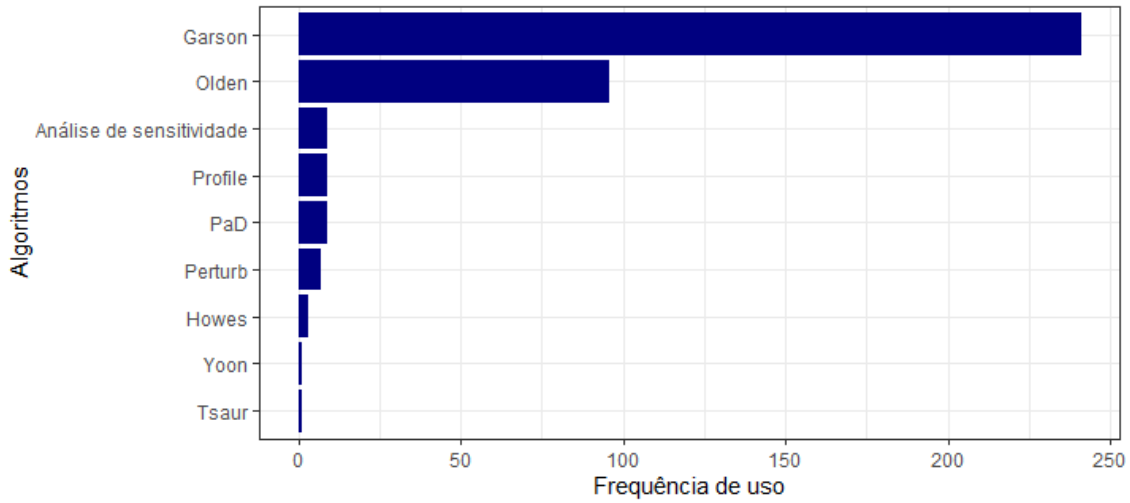


Figura 3.1: Frequência de uso dos algoritmos que quantificam a importância das variáveis de entrada em uma rede neural em artigos durante o período 2015 até outubro de 2020.

Os algoritmos de Howes, Yoon e Tsaur foram os menos utilizados desde 2015. Existem outras publicações que utilizaram estes algoritmos, porém, em data anterior a 2015, e portanto não foram incluídos no estudo apresentado neste capítulo. Os algoritmos de Garson e de Olden são os mais utilizados, sendo que o algoritmo de Garson possui mais que o dobro de uso em relação ao algoritmo de Olden, totalizando 77,5% dos artigos

selecionados. Trinta e oito dos 241 artigos que utilizaram o algoritmo de Garson também utilizaram outros algoritmos como Olden e os algoritmos de análise de sensitividade.

A Tabela 3.1 apresenta a quantidade de artigos que utilizaram WBFS por ano. Observa-se que há um crescente interesse pelo uso de tais técnicas, ocorrendo um aumento de 167% de publicações em 2020 em relação ao número de publicações em 2015. O número de publicações aumentou 55% no ano de 2016 em relação à 2015, e durante os anos de 2017-2019 o número de publicações se manteve estável (média $\pm$ desvio padrão, 56,6 $\pm$ 1,15).

Ano	Quantidade de publicações
2015	27
2016	42
2017	58
2018	56
2019	56
2020	72

Tabela 3.1: Quantidade de publicações que utilizaram os WBFS por ano.

A Figura 3.2 mostra os ocorrências anuais do uso de cada WBFS estudado nesta tese. Observa-se que desde 2015 o algoritmo de Garson é o mais utilizado, seguido pelo algoritmo de Olden. Os algoritmos de Tsaur e de Yoon foram utilizados apenas em 2017 e o algoritmo de Howes apenas em 2017 e 2018. A soma das frequências registradas nessa figura é maior do que os valores apresentados na Tabela 3.1, pois algumas publicações utilizaram mais de um WBFS no mesmo estudo.

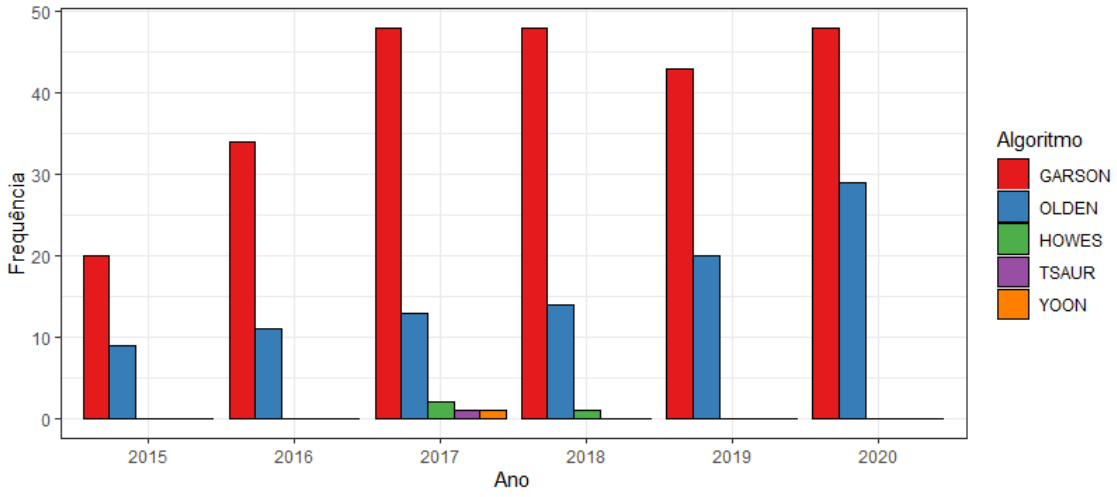


Figura 3.2: Ocorrências anuais do uso de cada WBFS nos artigos selecionados.

Foram identificadas 54 categorias de áreas de estudos diferentes nos artigos analisados. A Tabela 3.2 mostra a quantidade de artigos relacionados às 15 categorias de assunto mais frequentes. O campo "categoria de assunto" foi obtido a partir dos metadados extraídos do WoS. As publicações pertencem a diferentes áreas do conhecimento, sendo que as três principais são: engenharia, ciências ambientais e ecologia, e química. Este resultado confirma que os WBFS são utilizados como uma ferramenta para obter conhecimento em diferentes comunidades científicas, com predominância das ciências exatas e da terra, ciências biológicas e engenharias. A soma das ocorrências por categoria é maior do que o número de artigos revisados, pois existem estudos que se enquadraram em mais de uma categoria.

Categoria de assunto	Quantidade de ocorrências
Engenharia	137
Ciências ambientais e Ecologia	52
Química	36
Ciência da Computação	30
Ciência de Materiais	27
Energia, combustível	25
Recursos de água	22
Ciência, Tecnologia	21
Construção, Tecnologia de edifícios	17
Geologia	17
Termodinâmica	11
Agricultura	9
Transporte	8
Negócios, Economia	6
Mecânica	6

Tabela 3.2: Top 15 categorias dentre as categorias de assunto dos artigos revisados.

Dentre os artigos avaliados, apenas 5,15% apresentaram resultados relacionados a problemas de classificação, e os demais artigos apresentaram problemas de regressão. Além disso, 28% dos artigos afirmaram erroneamente que os WBFS são algoritmos de análise de sensitividade.

A Figura 3.3 mostra os padrões relacionados à quantidade de variáveis de entrada utilizadas nos estudos. Um estudo que utilizou uma base de dados com 400 variáveis foi omitido na construção da figura para facilitar a visualização do leitor. A grande maioria dos estudos se baseou em bases de dados com poucas amostras ($8{,}59 \pm 8{,}91$), e poucos estudos utilizaram conjunto de dados com mais de 20 variáveis.

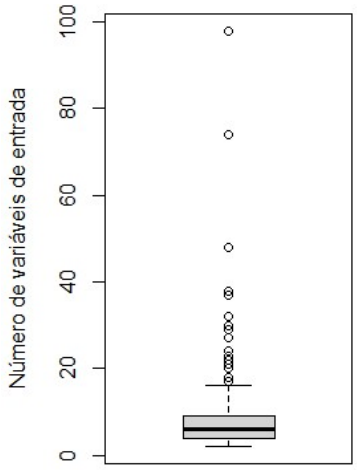


Figura 3.3: Boxplot do número de variáveis de entrada utilizadas nos artigos revisados.

3.3.2 Mineração de texto

As Figuras 3.4, 3.5, e 3.6 apresentam os dendrogramas relacionados às análises dos títulos, palavras-chave e resumos, respectivamente. Todas as sub figuras "a)" dessas imagens demonstram a análise de *cluster* hierárquico considerando todos os 311 documentos. Todas as sub figuras "b)" representam os *clusters* hierárquicos apenas das publicações relacionadas à Engenharia. Todas as sub figuras "c)" representam os *clusters* hierárquicos apenas das publicações da Ciências Ambientais e Ecologia. Todas as sub figuras "d)" representam os *clusters* hierárquicos apenas das publicações da Química. Todas as sub figuras "e)" representam os *clusters* hierárquicos das publicações da Ciência da Computação. Por fim, todas as sub figuras "f)" representam os *clusters* hierárquicos das publicações relacionadas à Ciência de Materiais. Essas áreas são as top-5 áreas de conhecimento apresentadas anteriormente na Tabela 3.2.

A quantidade de palavras que aparecem nos dendrogramas foram configuradas a partir de um parâmetro "*sparse*" na função que gerou os gráficos. Esse parâmetro é um número entre 0 e 1 que mede a porcentagem de zeros em cada termo e age como um ponto de corte. Um parâmetro *sparse* = 0,99 indica que são incluídos no gráfico todos os termos com quantidade $\leq$ 99% de zeros. Um parâmetro *sparse* = 0,1 indica que apenas os termos que possuem 10% ou menos de zeros serão mostrados. Nós ajustamos este termo de maneira a facilitar a leitura dos gráficos. Para as Figura 3.4 e 3.5 este parâmetro foi de 0,97. Por outro lado, a Figura 3.6 foi gerada com *sparse* = 0,7. Este ajuste foi necessário devido ao número de termos existentes nos documentos. A quantidade de palavras presente nos títulos dos artigos variam entre 15,2 $\pm$ 4,3, enquanto a quantidade média de palavras-chave é de 5,47, e a quantidade média de palavras nos resumos é de 209,66 $\pm$ 54,5. Considerando que a quantidade de palavras nos resumos é muito maior do que a quantidade de palavras presentes nos títulos e nas palavras-chave, a quantidade de

palavras no DTM dos resumos também foi superior e assim, o valor de *sparse* foi reduzido para diminuir a quantidade de palavras do dendrograma e facilitar a visualização do leitor.

As Figuras 3.4, 3.5, e 3.6 apresentam diferentes grupos que foram identificados por cores. O código que gerou as figuras foi configurado para colorir dois grupos com as cores cinza e vinho. Em seguida, analisamos cada figura e colorimos manualmente em azul os grupos de termos semelhantes específicos de cada área de estudo.

Considere a Figura 3.4 que apresenta os resultados em relação aos títulos dos artigos. É possível notar a existência de um grupo de palavras em todas as sub figuras formados por palavras relacionadas ao uso e aplicação de modelos de RNAs como "*artifici*", "*network*", "*neural*", "*model*", "*use*" e "*predict*". É importante ressaltar que este grupo de palavras foi definido ao solicitar que apenas dois grupos fossem identificados pelas cores cinza e vinho. Dessa forma, observa-se a caracterização de dois grupos gerais nas figuras. Grupo em vinho: termos relacionados ao uso de RNAs; e grupo em cinza: termos específicos dos estudos.

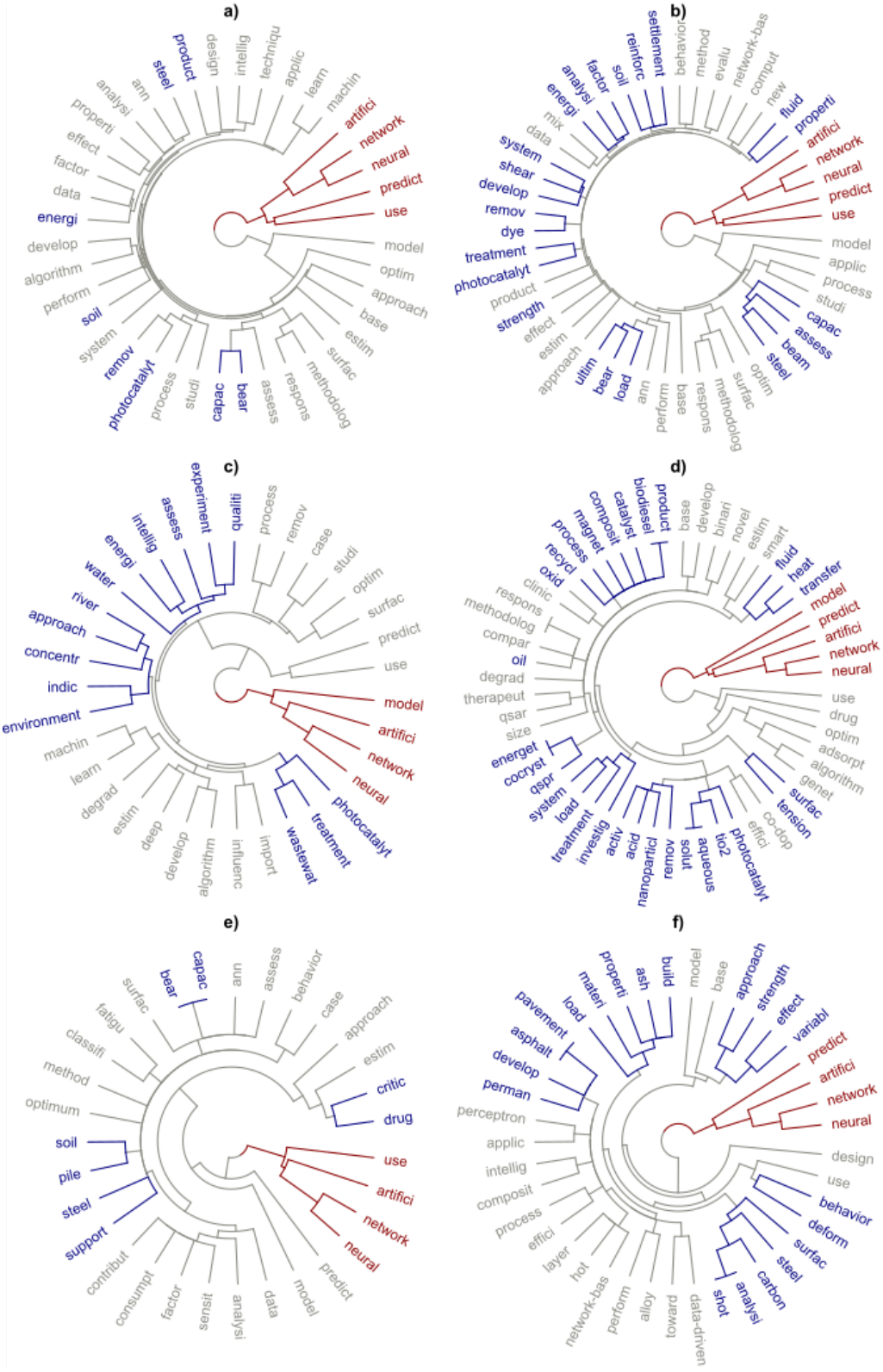


Figura 3.4: Dendrograma relacionado aos títulos das publicações.

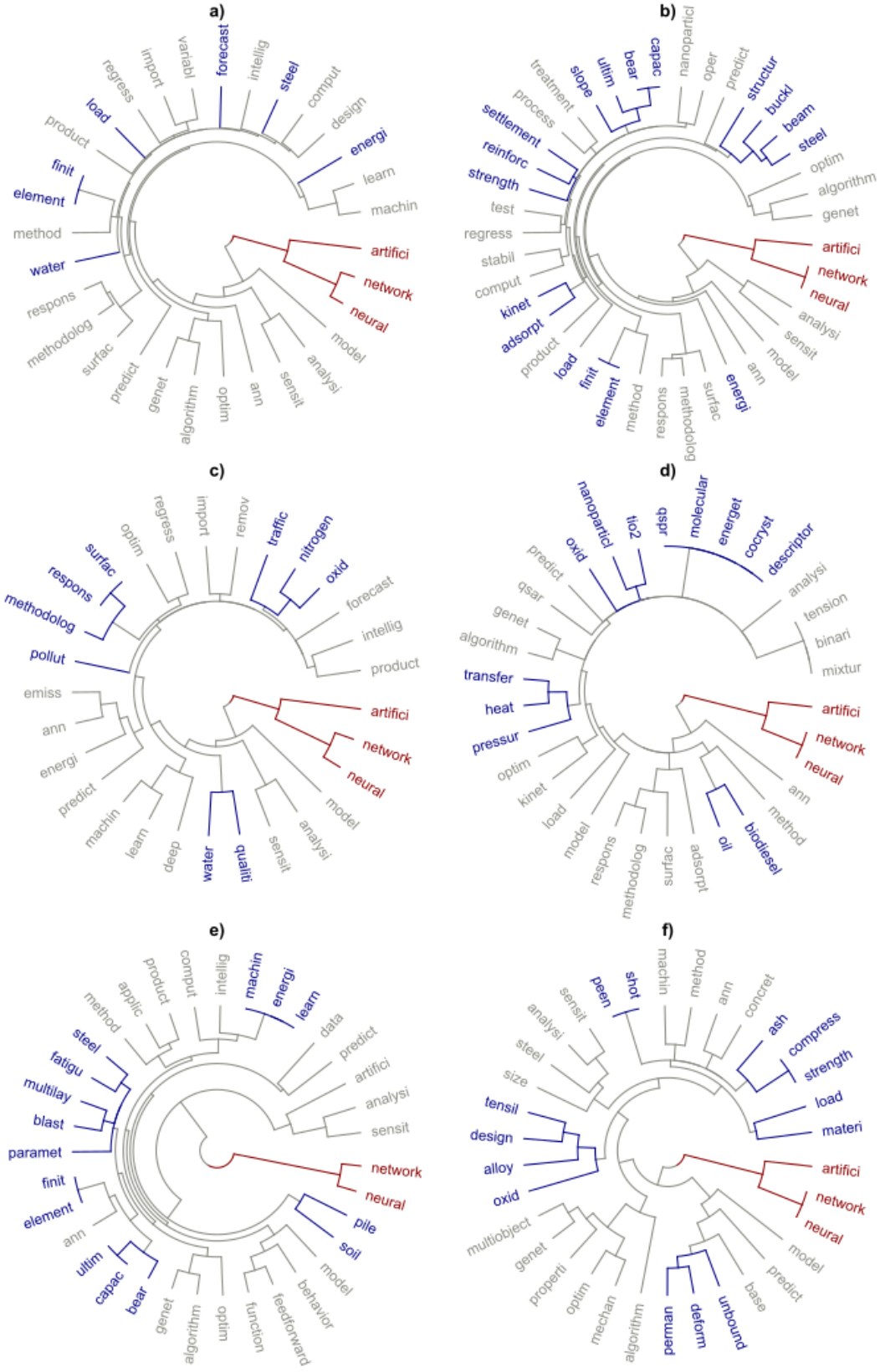


Figura 3.5: Dendrograma relacionado às palavras-chave das publicações.

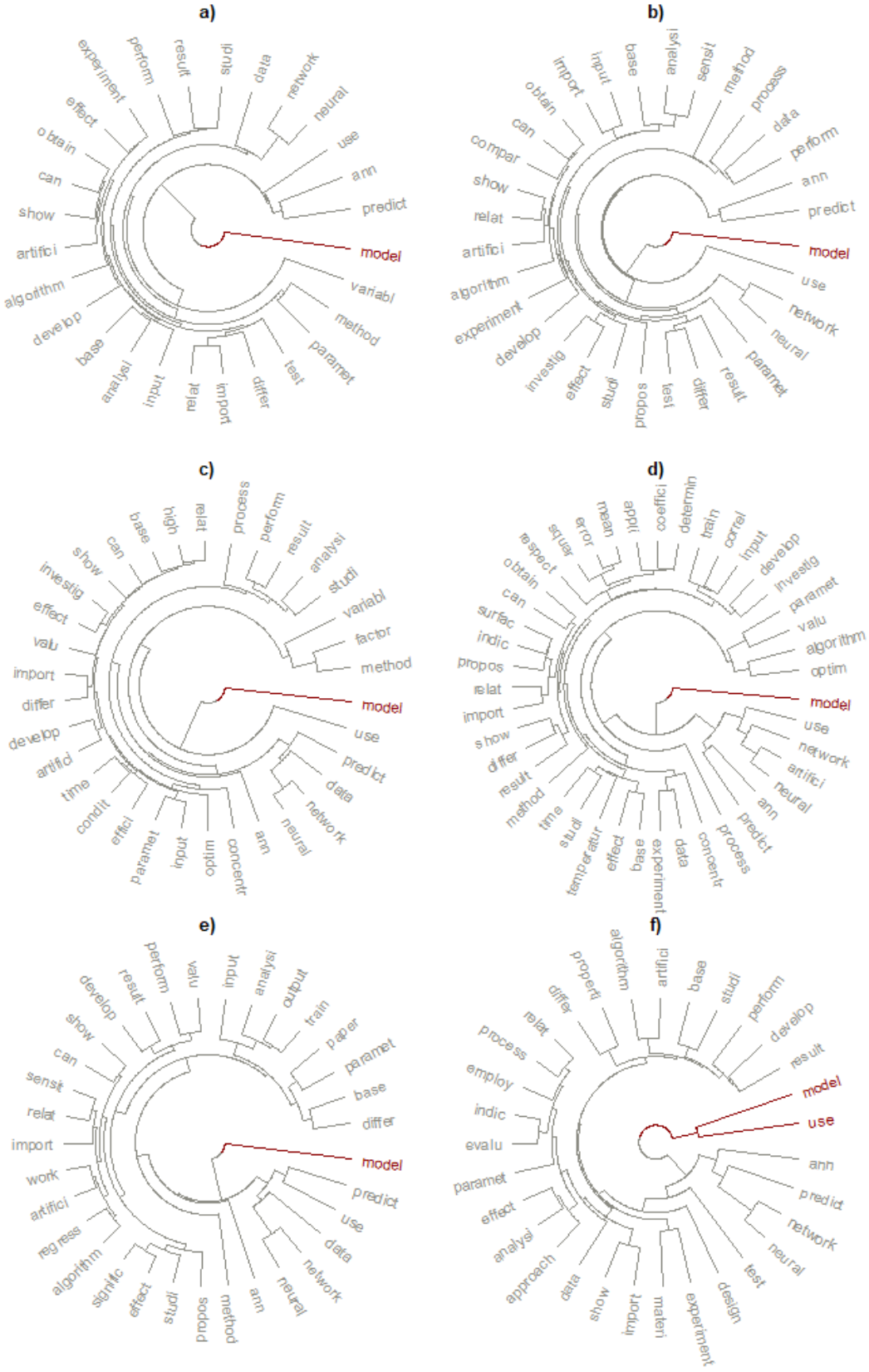


Figura 3.6: Dendrograma relacionado aos resumos das publicações.

Em seguida, o grupo em cinza foi analisado manualmente e os grupos de palavras específicos das áreas analisadas foram coloridos em azul. Observa-se que na sub figura 3.4 “a)” existem poucos grupos de termos específicos. A maioria dos termos estão relacionados à análise de dados e algoritmos. Isso pode ser justificado devido ao fato de existirem 54 áreas diferentes nos 311 artigos analisados. Dessa forma, a frequência maior de palavras esteve relacionada à aplicação de RNAs e análise de dados.

Por outro lado, ao analisar os artigos das top-5 áreas de estudo é possível identificar três ou mais grupos de palavras específicas contidas nos títulos das publicações. A maioria das palavras do grupo em cinza são relacionadas aos termos específicos das áreas, com poucos termos relacionados à análise de dados.

O mesmo padrão é observado ao analisar os dendrogramas das palavras-chave na Figura 3.5. De maneira similar à Figura 3.4, a sub figura 3.5 “a)” apresenta poucos termos específicos. As demais sub figuras 3.5 possuem termos característicos das áreas dos artigos semelhantes aos encontrados na Figura 3.4.

Por outro lado, os dendrogramas obtidos a partir dos resumos das publicações apresentam poucas informações sobre as áreas de estudo (Figura 3.6). Observe a divisão nos grupos cinza e vinho dessa figura. Em vinho obtivemos apenas a palavra “*model*”, enquanto outras palavras estão no grupo em cinza. Ademais, as palavras presentes nos dendrogramas estão relacionados a aspectos gerais de uma análise de dados e não nos dizem nada em específico.

De maneira geral, as top-5 categorias de estudos estão relacionadas a aplicações envolvendo os seguintes tópicos:

- **Engenharia:** inclinação, capacidade de carga, força, reforço, assentamento, estruturas, vigas de aço, carga, corante e fotocatalisador.
- **Ciências Ambientais e Ecologia:** água, rio, ambiente, esgoto, tratamento, óxido de nitrogênio, poluentes.
- **Química:** biodiesel, composição, óxido, óleo, co-cristal energético, fluidos, calor, energia, tratamento, ácido, solução aquosa, partícula e molécula.
- **Ciência da Computação:** viga de aço, solo, capacidade de carga, drogas, parâmetros, aprendizado de máquina, inteligência, computação, algoritmos genéticos, funções, otimização, modelos, análise de sensitividade e predição.
- **Ciência de Materiais:** projeto de tração, liga, óxido, material de carga, compressão, força, cinzas, carbono, aço, comportamento, deformação, asfalto e pavimentos.

3.3.3 Diferentes usos dos WBFS

A Figura 3.7 apresenta as três maneiras de se utilizar um WBFS que foram identificadas a partir da leitura dos artigos. Essas maneiras foram organizadas em três

categorias nomeadas A, B e C. Mais de 81% (252) dos artigos revisados utilizaram os WBFS conforme a categoria A, ou seja, treinaram apenas uma RNA e logo em seguida aplicaram o WBFS para estimar a importância das variáveis. Alguns estudos relataram que alguns testes foram realizados para encontrar a melhor arquitetura da RNA, mas não relatam a performance dos modelos gerados. Esses estudos foram designados na categoria de uso A, visto que não apresentaram informações suficientes sobre o conjunto de RNAs que foram treinadas para que nós analisássemos.

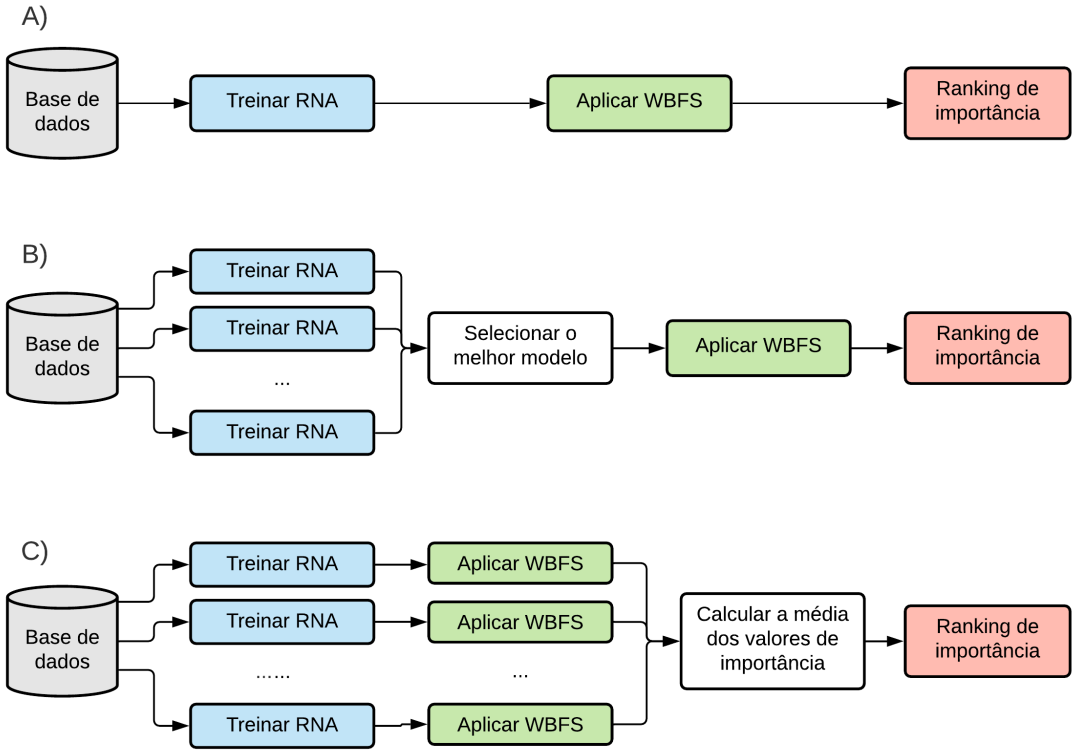


Figura 3.7: Diferentes categorias de uso dos WBFS.

A categoria B se baseia em treinar diferentes modelos de rede neural, selecionar o melhor modelo com base em sua capacidade de predição e aplicar o WBFS no modelo selecionado. A Tabela A.1 no Apêndice A apresenta os 38 artigos (12,21%) que se basearam na categoria B de uso dos WBFS, descrevendo o objetivo do estudo, a quantidade de RNAs treinadas e a especificação dessas redes. Observa-se que o número de modelos treinados é variável. A maioria dos estudos explora o número de neurônios, seguido pelos algoritmos de treinamento e funções de ativação. Na tabela NO se refere à "neurônios ocultos", LM se refere ao algoritmo de *Levenberg-Marquardt*, BP se refere a *backpropagation* e IBP se refere à *incremnetal backpropagation*, e BRNN se refere a RNA com *Bayesian Regularization*.

Por fim, a categoria de uso C se baseia em treinar diferentes modelos de rede neural e aplicar o WBFS em cada modelo gerado. Em seguida, o ranking de importância é calculado com base na média dos valores de importância obtidos em cada modelo. Essa abordagem foi sugerida por alguns autores como uma forma de lidar com a variação obtida nos rankings de importância devido à inicialização e atualização dos pesos de conexão [27, 97].

A Tabela A.2 no Apêndice A apresenta os 21 artigos (6,75%) que se basearam na categoria C de uso dos WBFS. Observa-se que não existem um padrão na quantidade de RNAs utilizadas para calcular a importância média das variáveis. Alguns estudos utilizam pouquíssimos modelos (quatro ou cinco RNAs) enquanto outros utilizam mais de 1000 RNAs como base para os WBFS. Alguns estudos utilizam a mesma arquitetura no conjunto RNAs e outros utilizam um conjunto de RNAs com diferentes arquiteturas. Mais estudos são necessários para estimar uma quantidade de modelos a serem utilizados, assim como avaliar se existe diferença ao utilizar RNAs com a mesma arquitetura e RNAs de arquiteturas diferentes.

3.3.4 Caracterização das RNAs

Diferentes tipos e arquiteturas de RNAs foram utilizadas nos 311 artigos revisados. A maioria dos estudos utilizaram uma rede neural de uma única camada oculta. Alguns estudos utilizaram mais de uma camada oculta na implementação da rede neural, caracterizando um estudo de *deep learning*. Em todos os estudos que utilizaram RNAs com múltiplas camadas ocultas o algoritmo de Olden foi utilizado. Isso é explicado devido ao fato de que apenas esse algoritmo pode ser utilizado em múltiplas camadas ocultas, ao realizar a soma dos produtos dos pesos de conexão entre as camadas, considerando o caminho entre o neurônio de entrada e o neurônio de saída.

Recentemente, Zhang et al. [158] utilizaram uma rede neural com arquitetura 5-25-20-18-15-1 para estimar o custo de capital de projetos de mineração. Os autores utilizaram o algoritmo de colonização de formigas para otimizar os pesos das RNAs e treinaram dez modelos com diferentes números de camadas ocultas. A rede com melhor desempenho foi com 4 camadas ocultas. O algoritmo de Olden foi utilizado para estimar a importância das variáveis com base em apenas um modelo de rede neural.

Bui e colaboradores [13] utilizaram diferentes modelos de aprendizado de máquina para predizer a sobrepressão de ar induzida pela explosão em mina a céu aberto. Os modelos utilizados foram: *Random Forest* (RF), Máquina Vetores de Suporte (SVM), Processo Gaussiano, árvores de regressão aditiva Bayesiana, árvores de regressão impulsionadas, k-vizinhos mais próximos (KNN), e oito RNAs com 1, 2 e 3 camadas ocultas. O modelo de RNA com arquitetura 7-9-7-5-1 obteve a melhor performance e foi utilizado

como base para o algoritmo de Olden.

Em outro estudo recente, Zhang et al. [162] utilizou uma rede neural com duas camadas ocultas para predizer a estabilidade de estradas em túneis subterrâneos. Os autores utilizaram diferentes algoritmos de aprendizado: SVM, sistema híbrido de inferência neural difusa (HYFIS), regressão linear múltipla, árvore de classificação e regressão, árvore de inferência condicional, e uma rede neural artificial com *particle swarm optimization* (PSO). O modelo de rede neural obteve o melhor desempenho. A importância das variáveis foi determinada utilizando o algoritmo de Olden com base em uma rede neural com arquitetura 9-12-17-1.

Por fim, o estudo de Manzanerez et al. [85] utilizou uma rede neural com múltiplas camadas e algoritmos genéticos para predizer a regulação da expressão da proteína Smad7 em pacientes com câncer de mama. A arquitetura da rede consistiu de 44 microRNAs (micro ácido ribonucleico) como neurônios de entrada, duas camadas ocultas com 42 e 63 neurônios, e uma camada de saída (44-42-63-1). Os autores utilizaram o algoritmo de Olden para estimar a importância de cada microRNA. Os valores reportados foram calculados a partir da média das importâncias de Olden obtidas em 20 treinamentos da rede neural, ou seja, este estudo utilizou da categoria C de uso dos WBFS explicada na seção anterior.

Além do uso de RNAs com uma ou múltiplas camadas ocultas, também observamos diferentes algoritmos de treinamento além do *backpropagration* com gradiente descendente, diferentes funções de ativação, e diferentes tipos de RNAs. Cinquenta e seis artigos utilizaram o algoritmo de LM para treinar as redes, 9 artigos utilizaram regulação Bayesiana, 5 artigos utilizaram PSO, 4 artigos utilizaram *Resilient Backpropagation* (RPROP) e um artigo utilizou BFGS (*Broyden-Fletcher-Goldfarb-Shanno algorithm*).

Uma parcela dos estudos avaliados (27%) não falou claramente qual foi a função de ativação utilizada. Entre os artigos que expressaram claramente a função de ativação usada, temos as funções sigmoide, tangente sigmoide (*tansig*), linear, função de base radial (RBF), ReLu e softmax. A função mais utilizada é a sigmoide, seguida pela *tansig* e a função linear. A frequência de uso relatada na Tabela 3.3 se refere ao uso das funções na camada oculta e na camada de saída. Diversos artigos utilizaram funções diferentes para cada camada.

Tabela 3.3: Frequência de uso de diferentes funções de ativação na camada oculta e na camada de saída das RNAs.

Função de ativação	Frequência de uso
sigmoid	214
tansig	117
linear	110
RBF	6
ReLu	6
softmax	1

Por fim, além de RNAs com uma camada oculta (MLP), ou com múltiplas camadas ocultas (*deep learning*), identificamos estudos que utilizaram ELM [141] e RNAs recorrentes [10].

3.4 Discussão

Os resultados do estudo apresentado neste capítulo caracterizam o uso e o contexto de aplicação dos WBFS. Todos 311 estudos revisados utilizaram algum WBFS para avaliar a importância das variáveis após o treinamento do modelo de classificação ou de regressão, de maneira a realizar inferências sobre o modelo obtido. A quantidade média de variáveis de entrada utilizadas nestes estudos facilita esta ação, visto que ela foi de 8,6. Avaliar o grau de importância de maneira manual em um conjunto de dados com poucas variáveis é uma tarefa realizável. No entanto, a medida que o número de variáveis aumenta, o grau de complexidade de compreender e encontrar justificativas para os níveis de importância de todas as variáveis também aumenta.

O uso dos WBFS em diferentes áreas de pesquisa demonstra o quanto estes métodos são difundidos. Apesar disso, um número considerável de estudos caracterizou os algoritmos de WBFS como análise de sensitividade. Esta é uma classificação errada visto que o objetivo da análise de sensitividade é diferente do objetivo dos WBFS. A análise de sensitividade visa observar a mudança na saída da rede por meio da alteração dos valores das variáveis. Os exemplos mais comuns de tais algoritmos são o método das derivadas parciais (PaD), que usa as derivadas parciais da saída em relação à uma entrada específica; o método *Profile*, que varia sucessivamente uma variável de entrada enquanto as outras são mantidas constantes em um valor fixo; e o método *Perturb*, que pode alterar as variáveis de entrada em diferentes faixas de valores. Por outro lado, os WBFS utilizam os pesos de conexão da rede neural para estimar a importância do(s) neurônio(s) de entrada em relação ao(s) neurônio(s) de saída da rede.

Diferentes tipos de RNA, número de neurônios e camadas ocultas, funções de ativação e algoritmos de treinamento foram observados nos estudos revisados. Alguns desses estudos demonstram a diferença na capacidade de predição do modelo a depender da arquitetura e/ou algoritmo de treinamento utilizado [124, 89, 56, 131, 121]

As diferenças na capacidade de predição de um modelo a depender da arquitetura e tipo de rede neural utilizada já é algo bem estabelecido na literatura. No entanto, não há um padrão universal que determina que uma arquitetura ou algoritmo será melhor para a solução de todo e qualquer problema, de maneira que se faz necessário realizar tentativas de diferentes arquiteturas para melhorar o modelo e, consequentemente, melhorar a solução do problema.

Considerando que diferentes arquiteturas e algoritmos levam a diferentes capacidades de predição, os valores da matriz de pesos resultante de cada rede neural também serão diferentes. Até onde sabemos, não existe um estudo que avalia as diferenças entre rankings de importância calculados por WBFS com base em diferentes arquiteturas e algoritmos de RNAs. O Capítulo 5 explora essa hipótese por meio de uma série de experimentos para comparar o resultado de WBFS em dois tipos de RNAs (MLP e ELM), cada modelo de RNA com diferentes arquiteturas.

3.4.1 Tratamento dos valores de importância em múltiplos neurônios na camada de saída

A quantidade de neurônios na camada de saída é um fator importante a ser considerado. Em modelos de regressão existe apenas um neurônio de saída na rede neural que vai determinar o valor numérico da classe de dados. O mesmo vale para problemas de classificação binários. Nesses casos, o valor de k nas equações que estabelecem os WBFS estudados nesta tese é 1 (Equações 2-15, 2-16, 2-17, 2-18, e 2-19, discutidas no Capítulo 2). Por outro lado, em modelos de classificação com mais de 2 classes, existe um neurônio de saída para cada classe de dados. Nestes casos, os WBFS geram um valor de importância para cada classe de dados. Dessa forma é necessário avaliar a importância das variáveis para cada classe de maneira individual, ou calcular a média das importâncias dos neurônios de saída pra cada variável de entrada.

Ha e colaboradores [51] utilizaram uma abordagem diferente para lidar com os múltiplos neurônios da camada de saída de uma rede neural para determinar os padrões de propriedade de veículos domésticos. A rede neural foi baseada em 10 variáveis de entrada, 19 neurônios na camada oculta, e 4 neurônios de saída que determinavam se a pessoa não tinha nenhum veículo, se a pessoa tinha uma moto, duas motos, ou um carro com ou sem o adicional de uma moto. A importância das variáveis foi calculada com base

no algoritmo de Olden, em que a importância final foi determinada pela soma dos valores absolutos obtidos para cada neurônio de saída.

O estudo de Zeroual e colaborados [156] também lidou com quatro neurônios na camada de saída para predizer a estabilidade estática e sísmica de pequenas barragens de terra homogêneas. A saída do modelo mapeou 4 parâmetros de segurança, enquanto a entrada do modelo se baseou em 11 variáveis. Os autores utilizaram os algoritmos de Garson e de Olden para estimar a importância das variáveis. No entanto, os autores relatam um valor de importância para cada variável de entrada e não descrevem como a importância obtida por cada neurônio de saída foi reduzida em um único valor.

Korteby e colaboradores [68] utilizaram uma RNA para caracterizar o mecanismo de crescimento de uma granulação fundida em leito de fluido *in situ*. O modelo desenvolvido pelos autores com arquitetura 3-5-5 foi utilizado como entrada para o algoritmo de Garson. No entanto, os autores relatam um valor de importância para cada variável de entrada e não descrevem como a importância obtida por cada um dos 5 neurônios de saída foi reduzida em um único valor.

Saurabh e colaboradores [118] avaliaram a relação entre o humor expressado em redes sociais e o preço de ações no Reino Unido por meio de uma rede neural de arquitetura 6-9-2-3. Cada neurônio de saída representou o preço das ações em baixo, médio e alto. Os autores reportam o valor de importância de cada uma das variáveis de entrada, porém, não é descrito como ou se a importância de cada neurônio foi mapeada em um único valor. Nesse estudo em particular, a avaliação das variáveis de entrada que mapearam as emoções raiva, medo, feliz, amor, triste e estresse em relação a cada classe de dados (preço das ações baixo, médio ou alto) poderia ser relevante para possivelmente associar diferentes humores com cada classe de dados.

Jbari e colaborados [61] utilizaram uma RNA para predizer os valores de concentração de permeado (Cp) e a concentração de retentado (Cr) do processo de tratamento de águas residuais. Os autores treinaram 20 RNAs com diferentes números de neurônios ocultos ($i = 1, 2, \cdots, 20$). A arquitetura do melhor modelo obtido foi de 5-3-2. Os autores utilizaram o algoritmo de Garson para estimar a importância das variáveis. Os valores de importância foram calculados para cada neurônio de saída e apresentados individualmente, considerando que cada saída da rede mapeou um valor quantitativo diferente (Cp e Cr).

De maneira similar ao estudo Jbari et al., Wang e colaborados [143] utilizaram uma rede neural de arquitetura 4-6-2 para predizer a inclinação e a interceptação da linha reta de comportamento de fluxo não linear de meios porosos. Os autores relataram a importância das variáveis para cada um dos dois neurônio da camada de saída.

Diversos estudos utilizaram o pacote *NeuralNetTools* para aplicar o algoritmo de Garson ou de Olden em modelos de RNA com um ou mais neurônios na camada de saída

[1, 25, 48, 49, 80, 100, 62, 118, 82, 162, 158, 82, 147, 40, 88]. No entanto, conforme descrito na documentação desse pacote [9], a implementação desses algoritmos funciona apenas para um neurônio de saída, ou seja, apenas para problemas de regressão ou de classificação binária.

Para investigar este tópico nós realizamos um teste com o conjunto de dados clássico *iris* do UCI *Machine Learning Repository*, que possui três classes de dados, *setosa*, *virginica* e *versicolor*. Observamos que, ao utilizar uma RNA com arquitetura 4-5-3, o algoritmo retornou as importâncias das variáveis. No entanto, a importância obtida foi em relação ao primeiro neurônio de saída. A importância das variáveis de entrada em relação a cada neurônio da camada de saída foi obtida apenas ao manipular a matriz de pesos para ser fornecida para as funções que calculam a importância das variáveis pelos WBFS de Garson e de Olden. No entanto, a falta de aviso de que apenas a importância relacionada ao primeiro neurônio é calculada, sendo necessário manipular a matriz de pesos para obter os valores de importância em relação às outras saídas da RNA, pode levar à resultados parciais.

3.4.2 Diferentes categorias de uso dos WBFS

Três categorias de uso dos WBFS foram identificadas a partir da leitura dos artigos. A grande maioria dos estudos se baseou em apenas uma RNA para calcular os valores de importância das variáveis conforme a categoria A e B, que descrevem o uso dos WBFS em apenas um modelo de RNA. No entanto, alguns autores demonstraram que a inicialização dos pesos da rede e o treinamento desta, pode levar a diferentes valores de importância. Para resolver esse problema os autores sugerem que a avaliação das variáveis deve ser realizada por meio da média dos valores de importância obtidos por um conjunto de RNAs [27, 97].

Apenas 6,75% dos estudos aalisados utilizaram a média da importância das variáveis (categoria C de uso dos WBFS). As principais escolhas para a apresentação dos resultados da média da importância das variáveis foram: boxplot, função de densidade ou apenas pela média $\pm$ desvio padrão [36, 64, 123, 72, 107, 110, 32, 74, 79, 88, 75, 157, 97, 96, 37]. Algumas publicações utilizaram a categoria C mas não descreveram qual foi a variação observada.

Apesar de a categoria de uso C ser uma opção, aparentemente melhor em relação ao uso de apenas uma única RNA, para calcular a importância das variáveis por meio de um WBFS, nós acreditamos que essa não é a maneira adequada de lidar com os WBFS devido à possibilidade de uma mesma variável assumir diferentes posições de importância no ranking, a depender do treinamento da rede.

O estudo de Schmitt e colaboradores [119] foi o único dos estudos selecionados

neste capítulo que avaliou a ordem de importância das variáveis em vez do valor de importância. Os autores treinaram 10 RNAs de mesma arquitetura (4-5-1) para prever a incrustação de membranas em um biorreator de membrana anóxico-aeróbica para tratar águas residuais domésticas. Os autores avaliaram a capacidade de predição e o ranking de importância obtido pelo algoritmo de Garson em cada um dos 10 modelos de RNA. Apesar de algumas redes apresentarem capacidades de predição semelhantes, a ordem de importância das variáveis nunca foi a mesma. Mediante esse resultado, os autores seguiram seu estudo sem considerar os resultados obtidos pelo algoritmo de Garson.

Até o presente momento, o estudo de Schmitt e colaboradores foi o único a avaliar as posições de importância obtidas por diferentes RNAs e apontar este problema. No entanto, não há nenhum estudo que avaliou sistematicamente se os valores de importância de várias RNAs treinadas levam à mesma ordem de importância ou se as diferenças entre os valores de importância podem causar alternância na posição de importância da variável.

3.5 Considerações finais

Este capítulo demonstrou as diferentes áreas e formas de uso de um WBFS. Após uma análise dos artigos publicados de 2015 até outubro de 2020, identificamos que a maioria das publicações são de áreas da engenharia, ciências exatas e da terra, e de ciências biológicas. A maior parte dos estudos utilizou um conjunto de dados de baixa dimensionalidade (média de variáveis igual a 8,6) e utilizaram os algoritmos de Garson e de Olden.

A maioria dos estudos se baseou em apenas uma RNA para calcular a importância das variáveis. Identificamos alguns estudos (6,75%) que utilizaram a média da importância das variáveis obtida por um conjunto de RNAs, as quais variaram de 4 até 10000 modelos. Esses estudos alegam que uma única RNA pode levar à resultados errôneos visto que o valor de importância das variáveis muda conforme os valores dos pesos de conexão são inicializados e treinados.

Motivados pela possibilidade de diferentes RNAs treinadas com base no mesmo conjunto de dados resultarem em diferentes rankings de importância, o próximo capítulo apresenta uma proposta de uma abordagem de votação para estimar a estabilidade dos resultados obtidos por um WBFS.

CAPÍTULO 4

Nova abordagem de avaliação dos WBFS

Neste capítulo apresentaremos a proposta da nova abordagem para avaliar os resultados de um WBFS. Este estudo foi publicado no *Journal Expert Systems with Applications* [21].

4.1 Motivação

Conforme abordado no capítulo anterior, alguns autores recomendam utilizar um conjunto de modelos de RNAs para calcular a importância das variáveis a partir de um WBFS [27, 97]. A recomendação ocorre em razão da variação obtida nos pesos de conexão das RNAs devido à inicialização aleatória seguida pelas atualizações realizadas nos mesmos durante o treinamento da RNA. No entanto, a maioria das aplicações que calcularam a importância das variáveis a partir de um WBFS desde 2015 se basearam em apenas uma RNA.

Mediante o exposto, este capítulo verifica a hipótese de que os rankings de importância gerados por um WBFS a partir de t RNAs treinadas além de diferir nos valores de importância, também diferem nas posições de rankings de importância das variáveis envolvidas.

A Definição 4.1 expressa a ideia sobre a posições dos rankings de importância, necessária para a compreensão da hipótese mencionada.

Definição 4.1. *Posição de importância. Dado o resultado de um seletor de variáveis que estabelece a importância de g variáveis por meio de um score, g posições de importâncias podem ser associadas aos scores. A variável que obteve o maior score no ranking é alocada na primeira posição de importância, a variável que obteve o segundo maior valor de importância é alocada na segunda posição, e assim sucessivamente até que a variável com o menor score seja alocada na g-ésima posição de importância.*

A partir dessa definição é possível investigar se uma variável é classificada em diferentes posições de importância por t rankings de importância. Caso a hipótese investigada neste capítulo seja corroborada, poderemos concluir que o uso da média dos

valores de importância obtidos por t RNAs talvez não seja a abordagem mais adequada para utilizar um WBFS, assim como calcular o WBFS com base em apenas um modelo de RNA pode levar a interpretações equivocadas.

Dessa maneira, este capítulo propõe uma nova metodologia para a avaliação de um WBFS baseado em uma abordagem de votação.

4.2 Proposta da abordagem de votação

A metodologia proposta foi inspirada no algoritmo *Random Forest* [12]. Este algoritmo utiliza de um conjunto de árvores de decisão para determinar o resultado de uma classificação ou de uma regressão, em que o resultado de cada árvore de decisão é tomada como um voto e o resultado final é definido pela classe mais votada.

A proposta da nova abordagem visa avaliar a estabilidade do resultado de um WBFS utilizando um mecanismo similar ao utilizado pelo *Random Forest*. A Definição 4.2 expressa a ideia sobre a estabilidade de um WBFS, necessária para a compreensão de nossa proposta.

Definição 4.2. *Estabilidade de um WBFS. O nível de estabilidade de um WBFS é calculado com base nas classificações obtidas nas diferentes posições de importância para cada variável do conjunto de dados analisado. Quanto maior a diferença entre as posições de importância das variáveis de um conjunto de dados geradas a partir de t modelos de redes neurais, maior é o grau de instabilidade do WBFS em questão.*

Considere um conjunto de t modelos de RNAs treinados a partir de um conjunto de dados que possui g variáveis. Cada modelo possui uma matriz de pesos que é utilizada para calcular a importância das variáveis a partir de um WBFS. Dessa maneira, cada modelo de RNA está relacionado a um vetor $\mathbf{u} = [u_1, u_2, \ldots, u_g]^T \in \mathbb{R}^g$ que armazena os valores de importância gerados por um método WBFS por meio dos pesos da RNA. Os $t \times \mathbf{u}$ vetores de importância são armazenados em uma matriz $\mathcal{U}$, conforme descrito a seguir.

$$\mathcal{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1g} \\ \vdots & \vdots & \ddots & \vdots \\ u_{t1} & u_{t2} & \cdots & u_{tg} \end{bmatrix}_{t \times g} \tag{4-1}$$

A partir da matriz $\mathcal{U}$ a abordagem de votação é aplicada, permitindo computar as métricas desenvolvidas para avaliar a estabilidade do WBFS, as quais chamaremos de métricas de estabilidade.

A abordagem de votação é composta pelos passos descritos a seguir:

1. Transformar os valores de importância $\mathbf{u}_i$ em posições de importância $\mathbf{p}_i$. O maior valor de importância em $\mathbf{u}_i$ recebe o valor de importância 1, o segundo maior valor recebe a posição de importância 2, até que o menor valor de importância receba a posição g.
2. Armazenar as posições de importância $\mathbf{p}$ em uma matriz

$$\mathcal{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1t} \\ \vdots & \vdots & \ddots & \vdots \\ p_{g1} & p_{g2} & \cdots & p_{gt} \end{bmatrix}_{g \times t} \tag{4-2}$$

3. Calcular uma matriz de contingência relacionando os votos que cada variável recebeu em cada posição de importância.
4. Calcular a confiança de cada variável, a confiança mínima, o número de posições de importância indicadas para cada variável, e a ordem de importância final das variáveis.

A abordagem de votação representa a quarta categoria de uso dos WBFS, categoria D, e é representada na Figura 4.1 e pelo Algoritmo 4.1.

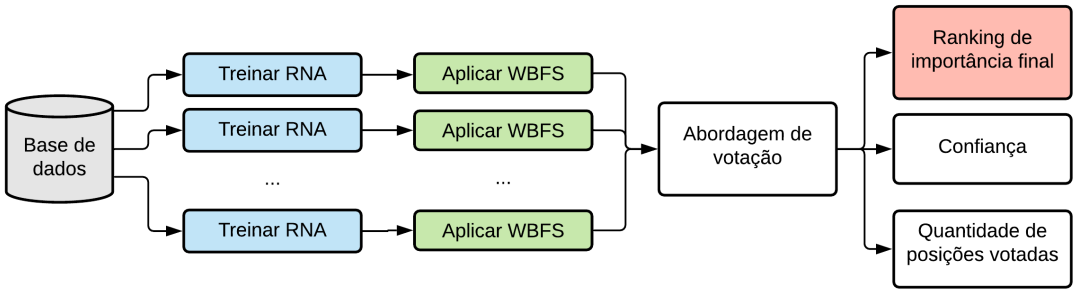


Figura 4.1: Proposta da abordagem de votação.

Algoritmo 4.1: Abordagem de Votação

Entrada: banco de dados $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\} | x \in \mathbb{R}^g$.

Saída: Ranking de importância, valores de confiança, quantidade de posições votadas, confiança mínima.

1 Treinar as RNAs com base nos dados de entrada.
2 Aplicar o WBFS na matriz de pesos de cada modelo RNA.
3 Transformar os valores de importância de cada ranking em posições de importância.
4 Criar uma matriz de contingência com o número de votos que cada variável recebeu em cada posição de importância.
5 Determinar o ranking final com base na posição de importância mais votada em cada variável.
6 Calcular o valor de confiança de cada variável.
7 Calcular o valor de confiança mínimo.
8 Calcular o número de diferentes posições que uma variável foi votada.

Considere a aplicação de um WBFS em cinco modelos de RNAs treinados $(t = 5)$ em um conjunto de dados fictício composto de oito variáveis $(g = 8)$, resultando assim em um conjunto de valores de importância $\mathcal{U} = (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_5) | \mathbf{u} \in \mathbb{R}^g$. Os valores de importância $\mathcal{U}$ são transformados para as posições de importância $\mathcal{P} = (\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_5) | \mathbf{p} \in \mathbb{R}^g$. A matriz $\mathcal{P}$, contendo os elementos $p_{ij} | i = (1, ..., g)$ e $j = (1, \cdots, t)$ que representam a posição de importância obtidos pela j-ésima RNA para a i-ésima variável, pode ser representada da seguinte maneira para este exemplo:

$$\mathcal{P} = \begin{bmatrix} V1 & V1 & V6 & V4 & V1 \\ V4 & V4 & V1 & V1 & V4 \\ V2 & V6 & V2 & V2 & V2 \\ V6 & V2 & V4 & V6 & V6 \\ V3 & V3 & V3 & V3 & V3 \\ V8 & V8 & V5 & V8 & V8 \\ V7 & V5 & V8 & V5 & V5 \\ V5 & V7 & V7 & V7 & V7 \end{bmatrix}_{8 \times 5} \tag{4-3}$$

A matriz de importância $\mathcal{P}$ é utilizada para construir a tabela de contingência C (Tabela 4.1), que apresenta a frequência de votos que uma variável recebeu para cada posição de importância, os valores de confiança e o ranking de importância final.

Tabela 4.1: Tabela de contingência para os dados apresentados na Equação 4-3.

Ranking / Variáveis	1	2	3	4	5	6	7	8	Confiança
V1	**3**	2	-	-	-	-	-	-	0,6
V2	-	-	**4**	1	-	-	-	-	0,8
V3	-	-	-	-	**5**	-	-	-	1
V4	1	**3**	-	1	-	-	-	-	0,6
V5	-	-	-	-	-	1	**3**	1	0,6
V6	1	-	1	**3**	-	-	-	-	0,6
V7	-	-	-	-	-	-	1	**4**	0,8
V8	-	-	-	-	-	**4**	1	-	0,8
Ranking final	V1	V4	V2	V6	V3	V8	V5	V7	0,725

Os valores armazenados na tabela são utilizados para calcular o valor de confiança que mede a taxa de indicação de uma variável para uma determinada posição de importância, calculada pela Equação (4-4), em que i representa a i-ésima variável, tal que $i = (1,2,\cdots,g)$.

$$Confianca_i = \frac{max(C_i)}{t} \tag{4-4}$$

Os valores de confiança para o exemplo fictício são apresentados na última coluna da Tabela 4.1. A última linha da tabela apresenta o ranking final composto pelas posições mais votadas em cada variável. Para auxiliar na avaliação dos resultados da métrica confiança é necessário avaliar a confiança mínima. A confiança varia em um intervalo de β a 1, em que β é a confiança mínima, definida por pela Equação 4-5.

$$\beta = \frac{1}{g} \tag{4-5}$$

O valor mínimo da confiança ocorre quando todas as posições de importância recebem o mesmo número de votos. Quanto mais próximo de 1 é o valor de confiança, mais estável é a posição dada a uma variável, ou seja, um número próximo de t de RNAs concordam com a posição de importância da variável em questão.

A posição de importância final de uma variável é a posição mais votada para a variável em questão. Por consequência, o valor da confiança mínima auxilia na avaliação da confiança das posições de importância de cada variável pois atua como um limite mínimo. Os empates entre uma mesma posição de importância nos rankings individuais e no ranking final foram tratados de maneira aleatória entre as posições envolvidas.

Uma variável pode ser classificada em diferentes posições de importância a depender do quão discrepante os rankings de importância armazenados $\mathcal{U}$ são. Nesse contexto, ε define o número de diferentes posições que uma variável foi classificada ao longo de C_i. Esse valor pode variar em um intervalo de $1-g$, em que 1 indica que a variável foi votada para a mesma posição de importância por todos os rankings. O caso $\varepsilon = g$ indica que a variável foi classificada em todas as g posições de importância pelos rankings de importância. Quanto mais próximo de g for ε, mais instável será o resultado do WBFS para a variável relacionada.

4.3 Materiais e Métodos

A proposta da nova metodologia de avaliação foi utilizada para avaliar os algoritmos de Garson, Olden, Yoon, Tsaur, e de Howes e Crook, e vice versa. Nós utilizamos oito bases de dados simulados e 4 bases de dados públicas provindas do *UCI Machine Learning Repository*.

Quinhentas RNAs do tipo MLP com regularização por *weight decay* foram treinadas para cada base de dados. O conjunto de RNAs foi composto por 50 redes para cada número h de neurônios ocultos, em que $h = (3,4,5,\cdots,12)$, totalizando assim, 500 modelos de RNAs para cada base de dados.

Cada modelo de RNA foi treinado de maneira a otimizar o parâmetro *weight decay* por meio de uma busca em grade pelos valores 10^{-i}, em que $i = [1,2,...,5]$. Cada um dos parâmetros de *weight decay* foi utilizado durante o treinamento de cada uma das 500 redes neurais, de maneira que o modelo associado ao *weight decay* que resultava no maior desempenho foi escolhido. Essa otimização do parâmetro foi definida com o objetivo de utilizar os modelos de RNAs mais precisos.

Todas os modelos de RNA foram treinados de maneira independente e com a inicialização dos pesos de conexão em valores pequenos de forma aleatória, juntamente com uma validação cruzada de 10-*folds*.

4.3.1 Detalhes de implementação

Os códigos desenvolvidos foram escritos em R, uma plataforma de código aberto para a computação estatística e desenvolvimento de gráficos [104]. Além de funções rotineiras disponíveis no pacote *base*, diferentes pacotes disponíveis no R foram aproveitados nesta pesquisa.

O modelo de RNA utilizado foi proveniente do pacote *nnet*, modelado a partir do pacote *caret*, ambos disponíveis pelo repositório CRAN do R. O pacote *caret* possibilita o treinamento de diferentes modelos de classificação e regressão. Além disso, o pacote

disponibiliza ferramentas simples para definir opções de treinamento, como a otimização de parâmetros do modelo a ser treinado, assim como o particionamento dos dados para treino e teste.

O pacote *doParallel* foi utilizado em todas as funções em que um laço de repetição fosse executado. Este pacote disponibiliza um *backend* paralelo para o pacote *foreach*. Este último foi utilizado para implementar as repetições.

4.3.2 Bancos de dados

Dois grupos de bancos de dados foram utilizados para testar a abordagem de votação e, consequentemente, avaliar a estabilidade dos WBFS estudados. O primeiro grupo é composto por conjuntos de dados simulados, os quais foram elaborados por nós e possuem uma relação de importância/contribuição pré-estabelecida entre as variáveis. Esse recurso nos possibilita comparar a real importância das variáveis para a classe de dados com a importância gerada pelos WBFS.

Oito bancos de dados foram elaborados, sendo que quatro deles descrevem problemas de regressão e quatro descrevem problemas de classificação. Os problemas de regressão foram definidos com base em três vetores aleatórios normalmente distribuídos (a, b, e c) compostos por 500 amostras, com média e desvio padrão iguais a 1. Os três vetores independentes foram utilizados para gerar a variável dependente com base nas equações definidas no estudo conduzido por Fischer [35].

Quatro relações foram exploradas entre as variáveis, sendo elas a relação linear (Equação 4-6), relação positiva quadrática (Equação 4-7), relação negativa quadrática (Equação 4-8), e a relação interativa (Equação 4-9).

$$y = 1 \times a + 0,5 \times b + 0,1 \times c \tag{4-6}$$

$$y = 1 \times a^2 + 0,5 \times b + 0,1 \times c \tag{4-7}$$

$$y = -1 \times a^2 + 0,5 \times b + 0,1 \times c \tag{4-8}$$

$$y = a \times b + 0,1 \times c \tag{4-9}$$

Cada conjunto de dados de regressão foi elaborado com base em apenas uma dessas relações, sendo y a variável dependente. Os problemas de classificação foram definidos com base nas mesmas variáveis independentes ($a, b, c, n = 500$). A variável dependente w, que representa a classe de dados dos problemas de classificação, foi definida conforme a regra apresentada na Equação (4-10).

$$w = \begin{cases} 0, & \text{se } \sigma(y) > median(\sigma(y)) \\ 1, & \text{se } \sigma(y) \leq median(\sigma(y)) \end{cases} \tag{4-10}$$

A regra estabelecida na Equação (4-10) utiliza a função sigmoide, definida por σ para transformar os dados na faixa de valores entre 0 e 1 (Equação 4-11). A classe de dados foi definida a partir da mediana do resultado da função sigmoide. Dessa maneira, cada classe de dados foi composta por 250 amostras, caracterizando problemas de classificação balanceados.

$$\sigma(y) = \frac{1}{1 + e^{-y}} \tag{4-11}$$

Em resumo, cada conjunto de dados simulado foi composto das variáveis independentes a, b e c, e sua respectiva variável dependente y para problemas de regressão e w para os problemas de classificação. A Tabela 4.2 resume as características dos bancos de dados simulados e as equações utilizadas para gerar as variáveis dependentes de cada banco de dados.

Tabela 4.2: Descrição dos bancos de dados simulados.

Identificação do banco de dados	Característica do modelo	Equação utilizada para gerar a variável dependente
#R1	Regressão	4-6
#R2	Regressão	4-7
#R3	Regressão	4-8
#R4	Regressão	4-9
#C1	Classificação	4-6 e 4-10
#C2	Classificação	4-7 e 4-10
#C3	Classificação	4-8 e 4-10
#C4	Classificação	4-9 e 4-10

A Tabela 4.3 apresenta o coeficiente de correlação de *Pearson* entre as variáveis independentes e dependentes. Observa-se que, nos conjunto de dados #R1, #R2, #C1 e #C2, a ordem de contribuição das variáveis independentes para a variável dependente é $a > b > c$, conforme o esperado considerando as Equações (4-6) e (4-7). Para os conjuntos de dados #R3 e #C3 a correlação é negativa entre a variável a e a classe de dados, conforme o esperado, considerando a Equação (4-8). Por fim, a relação iterativa entre as variáveis a e b nos conjuntos de dados #R4 e #C4 é aproximada para ambas variáveis, conforme o esperado considerando a relação interativa entre as variáveis a e b na Equação (4-9).

Tabela 4.3: Correlação entre a variável dependente com as variáveis independentes dos conjuntos de dados simulados.

			Coeficiente de correlação de Pearson		
	ID		*a*	*b*	*c*
Problemas de regressão	*#R1*	*y*	0,9	0,43	0,12
	#R2	*y*	0,78	0,14	0,05
	#R3	*y*	-0,76	0,26	0,04
	#R4	*y*	0,53	0,56	0,06
Problemas de classificação	*#C1*	*w*	0,84	0,41	0,11
	#C2	*w*	0,66	0,33	0,1
	#C3	*w*	-0,68	0,41	0,09
	#C4	*w*	0,51	0,51	0,1

A correlação de Pearson das variáveis independentes com a variável dependente será utilizada como referência para avaliar se os WBFS são capazes de identificar a importância correta entre as variáveis.

Em relação aos conjuntos de dados reais, não há informação pré-existente sobre a importância das variáveis independentes em ralação às variáveis dependentes. Dessa forma, esses conjuntos de dados nos permitirão medir a estabilidade dos métodos mediante dados que a importância real das variáveis é desconhecida.

O *UCI Machine Learning Repository* é um repositório de conjuntos de dados que compreende muitos domínios. Quatro conjuntos deste repositório foram usados para os experimentos. Um resumo desses conjuntos de dados é fornecido na Tabela 4.4.

Tabela 4.4: Descrição dos conjuntos de dados do *UCI Machine Learning Repository*

Bases de dados	Número de amostras	Número de variáveis	Característica do modelo	Número de classes
Haberman	306	4	Classificação	2
Iris	150	5	Classificação	3
Parkinsons	195	23	Classificação	2
Wine Quality	6497	13	Classificação	2

4.3.3 Validação dos resultados

O teste estatístico de Friedman foi selecionado para avaliar os resultados obtidos em cada métrica de estabilidade em cada grupo de conjunto de dados analisados. O teste de Friedman é um método não paramétrico para comparação de múltiplos grupos de amostras pareadas. As hipóteses do teste são:

Hipótese Nula (H_0). *Não existe diferença significativa entre os resultados provindos de diferentes algoritmos.*

Hipótese Alternativa (H_1). *Existe diferença significativa entre os resultados provindos de diferentes algoritmos.*

Essas hipóteses foram testadas a partir dos resultados obtidos nas métricas de confiança e o número de diferentes posições que uma variável foi classificada (ε).

Para testar H_0, o teste de Friedman organiza um ranking de desempenho para cada banco de dados. A melhor performance recebe o ranking 1, a segunda melhor performance recebe o ranking 2, e assim sucessivamente. Em caso de empate, a média do ranking é atribuído a cada um dos algoritmos.

A estatística do teste de Friedman é calculada com base nos rankings médios dos algoritmos, conforme a seguir:

$$R_j = \frac{1}{N}\sum_i r_i^j \tag{4-12}$$

em que R_j é o ranking médio do j-ésimo dos k algoritmos, e r_i^j representa o ranking obtido no j-ésimo dos k algoritmos e o i-ésimo dos N conjuntos de dados. Sob a hipótese nula de que os algoritmos são equivalentes, os valores de R_j também devem ser equivalentes. A estatística de Friedman é distribuída de acordo com χ_F^2 com (k - 1) graus de liberdade (Equação 4-13).

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_i R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{4-13}$$

Caso a hipótese nula seja rejeitada é necessário realizar um teste *post-hoc* para identificar quais algoritmos possuem diferentes performances [30]. Utilizamos o teste de Nemenyi conforme sugerido por Demšar [30]. Esse teste inclui o cálculo da diferença crítica, uma métrica que é utilizada para comparar as diferenças entre as médias dos rankings. A Equação (4-14) mostra o cálculo da diferença crítica para o teste de Nemenyi.

$$DC = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{4-14}$$

em que q_α é o valor crítico no nível de confiança α, k é o número de diferentes algoritmos analisados e N é o número de conjunto de dados analisados.

O teste de Friedman também foi utilizado para verificar se existe diferença significativa entre os grupos de redes neurais. Dentre as 500 RNAs treinadas para cada conjunto de dados existem 10 grupos de 50 RNAs cada, sendo que as RNAs de um mesmo grupo possuem o mesmo número de neurônios na camada oculta.

4.4 Resultados

O primeiro aspecto a ser analisado nos experimentos numéricos foi se existe diferença entre os resultados obtidos nas diferentes arquiteturas de redes neurais. Ou seja, avaliamos se as 50 redes com $h \in (3, 4, \cdots, 12)$ neurônios ocultos resultaram em métricas de estabilidade significativamente diferentes.

As hipóteses nula e a hipótese alternativa para essa fase dos experimentos são:

Hipótese Nula (H_0). *Não existe diferença significativa nos resultados das métricas de estabilidade provindas de RNAs com diferentes arquiteturas.*

Hipótese Alternativa (H_1). *Existe diferença significativa nos resultados das métricas de estabilidade provindas de RNAs com diferentes arquiteturas.*

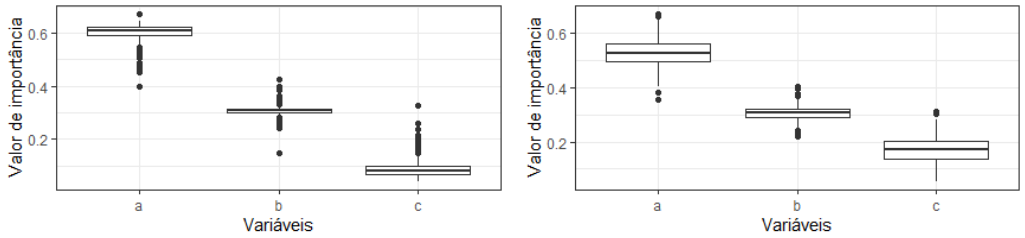
Esse foi o primeiro aspecto analisado visto que, caso existam diferenças significativas entre as arquiteturas das RNAs, as métricas de desempenho devem ser analisadas dentro do grupo de resultados de sua respectivas arquiteturas. Caso não exista diferença significativa, os resultados das métricas de desempenho podem ser analisadas com base em todo o conjunto de 500 RNAs.

Nenhuma diferença significativa foi encontrada entre o número de neurônios ocultos e os valores de confiança (valor $p = 0{,}9804$, $\chi^2 = 2.51$), ou com os valores ε (valor $p = 0{,}9966$, $\chi^2 = 1.56$). Dessa maneira, os resultados relatados a seguir são baseados nos rankings obtidos pelas 500 modelos de RNAs para cada banco de dados.

4.4.1 Experimentos numéricos com os dados simulados

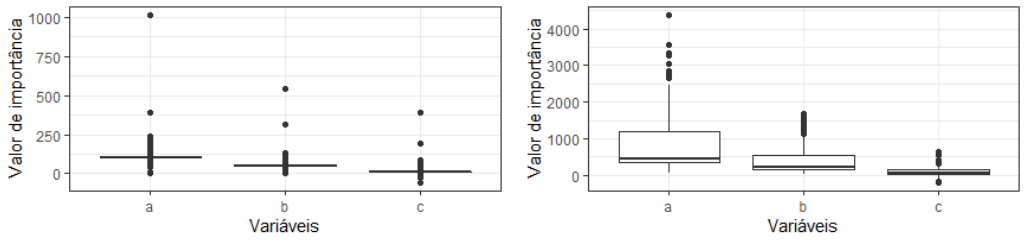
Caracterização das RNAs e dos valores de importância

Os rankings de importância obtidos com os algoritmos de Garson, Olden, Yoon, Tsaur e Howes e Crook resultaram em diferentes valores de importância entre as 500 redes neurais. Conforme esperado, todos os rankings de importância demonstraram uma alta variabilidade nos valores de importância para problemas de classificação e de regressão. As Figuras 4.2, 4.3, 4.4, e 4.5 apresentam os boxplots dos valores de importância obtidos para cada variável, em cada algoritmo WBFS, para os problemas de classificação e de regressão. Em alguns casos é possível observar a ordem de importância das variáveis, como na relação linear entre os dados para os algoritmos de Garson e Yoon (Figura 4.2). Entretanto, na maioria dos casos, os valores de importância de cada variável se sobrepõem, não sendo possível afirmar a ordem de importância com base apenas nesses gráficos.



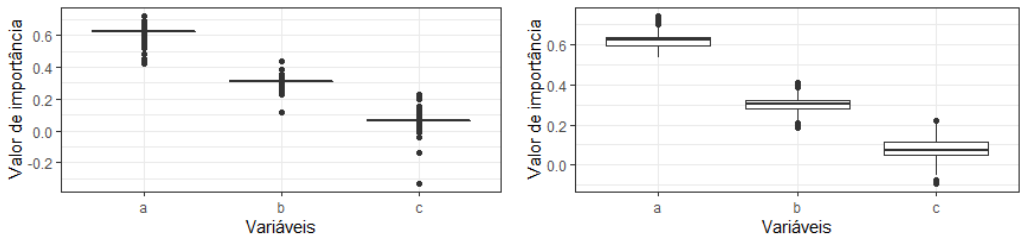
(a) #R1 - Garson

(b) #C1 - Garson



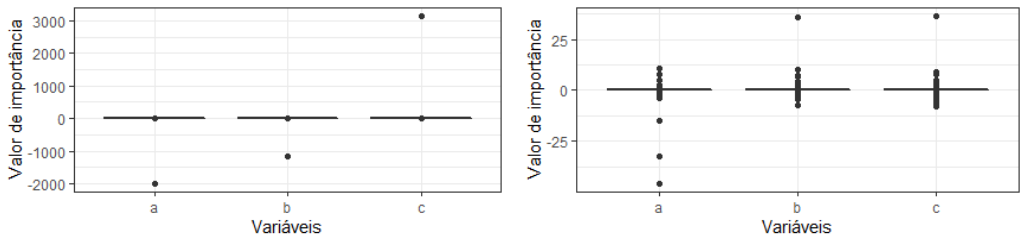
(c) #R1 - Olden

(d) #C1 - Olden



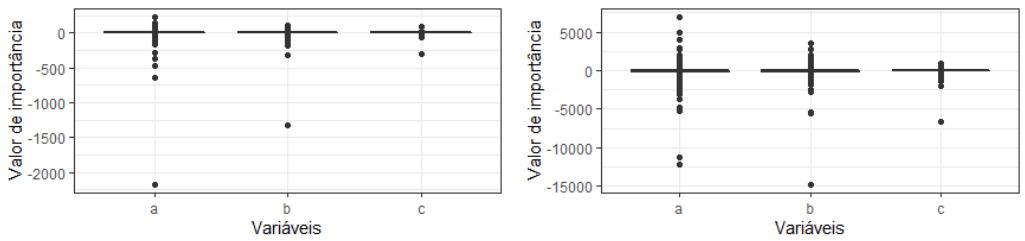
(e) #R1 - Yoon

(f) #C1 - Yoon



(g) #R1 - Tsaur

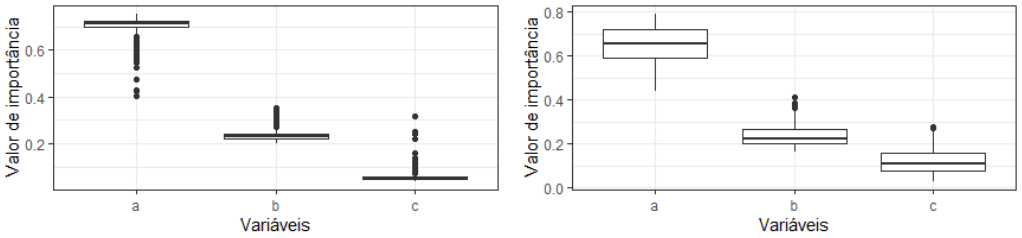
(h) #C1 - Tsaur



(i) #R1 - Howes

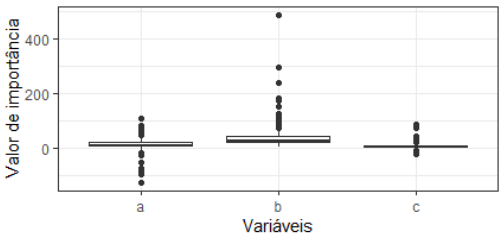
(j) #C1 - Howes

Figura 4.2: Boxplot dos valores de importância obtidos a partir das 500 RNAs conforme os algoritmos de Garson, Olden, Yoon, Tsaur, e de Howes e Crook para o conjunto de dados simulados #R1 e #C1.

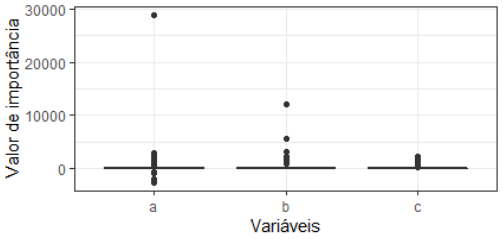


(a) #R2 - Garson

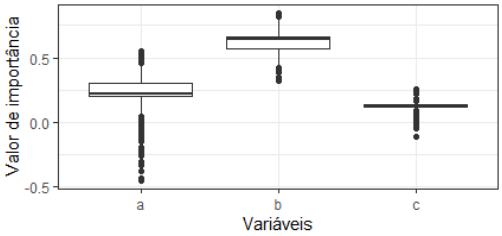
(b) #C2 - Garson



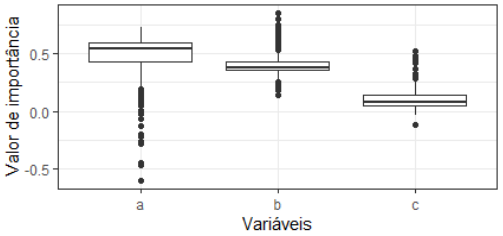
(c) #R2 - Olden



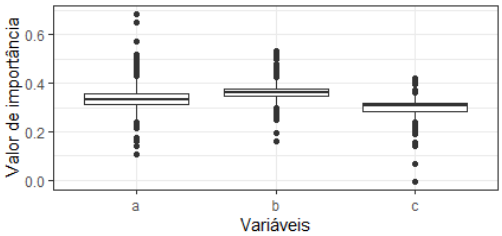
(d) #C2 - Olden



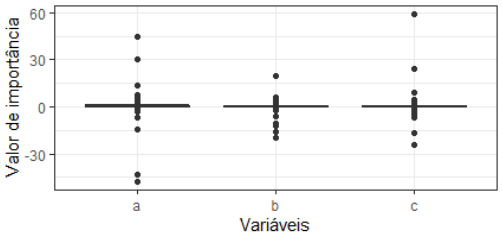
(e) #R2 - Yoon



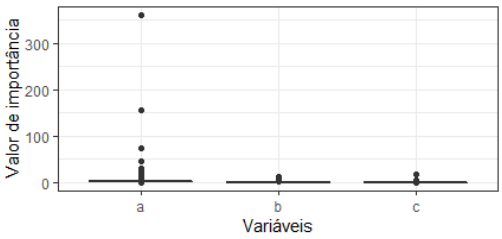
(f) #C2 - Yoon



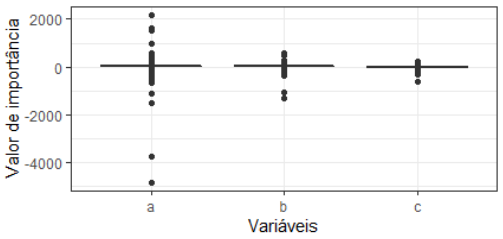
(g) #R2 - Tsaur



(h) #C2 - Tsaur

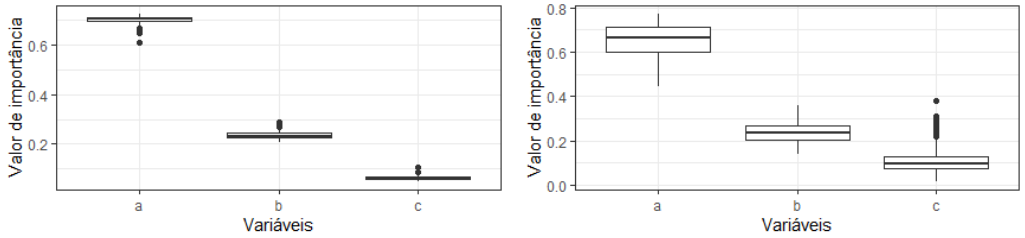


(i) #R2 - Howes



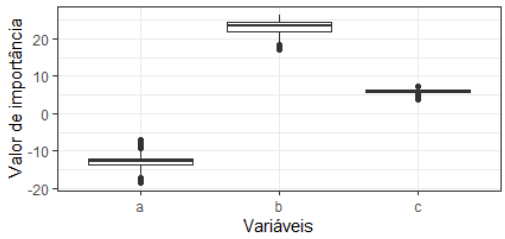
(j) #C2 - Howes

Figura 4.3: Boxplot dos valores de importância obtidos a partir das 500 RNAs conforme os algoritmos de Garson, Olden, Yoon, Tsaur, e de Howes e Crook para o conjunto de dados simulados #R2 e #C2.

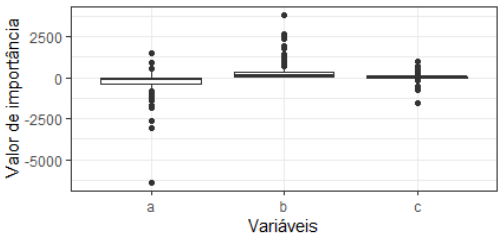


(a) #R3 - Garson

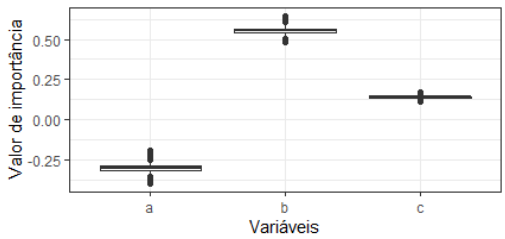
(b) #R3 - Garson



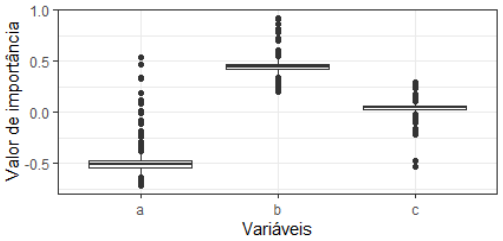
(c) #R3 - Olden



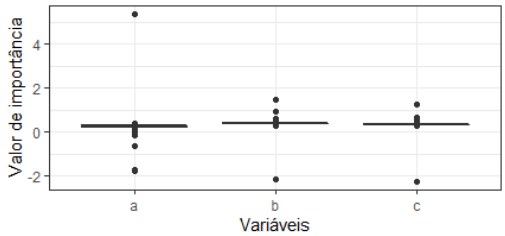
(d) #C3 - Olden



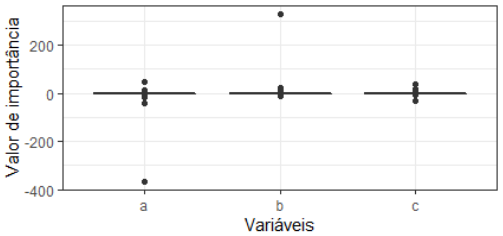
(e) #R3 - Yoon



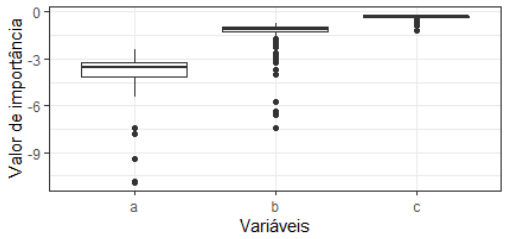
(f) #C3 - Yoon



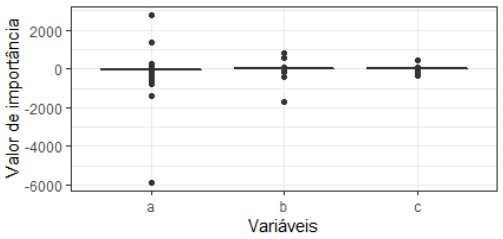
(g) #R3 - Tsaur



(h) #C3 - Tsaur

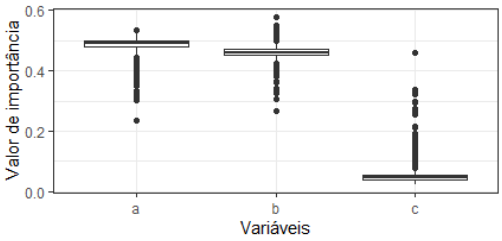


(i) #R3 - Howes

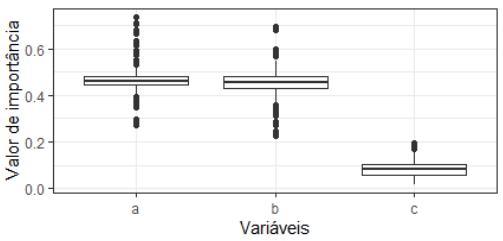


(j) #C3 - Howes

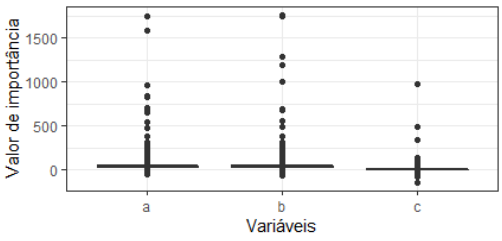
Figura 4.4: Boxplot dos valores de importância obtidos a partir das 500 RNAs conforme os algoritmos de Garson, Olden, Yoon, Tsaur, e de Howes e Crook para o conjunto de dados simulados #R3 e #C3.



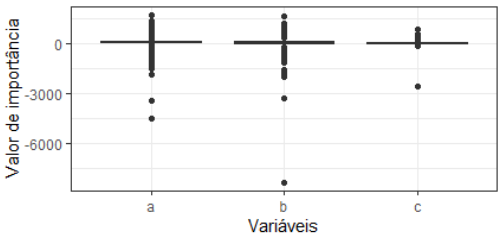
(a) #R4 - Garson



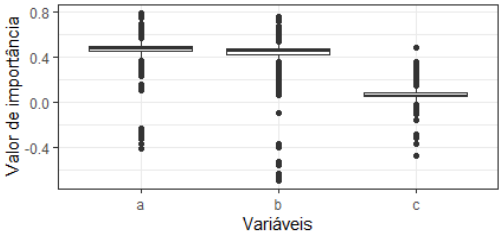
(b) #R4 - Garson



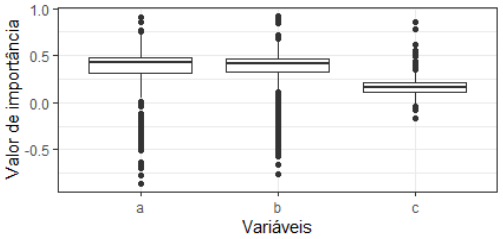
(c) #R4 - Olden



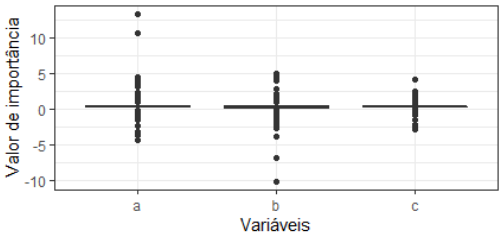
(d) #C4 - Olden



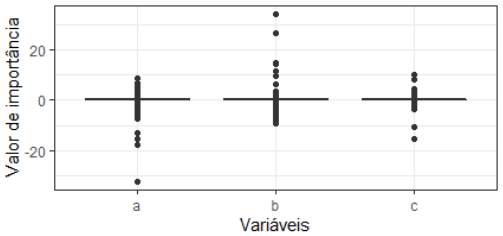
(e) #R4 - Yoon



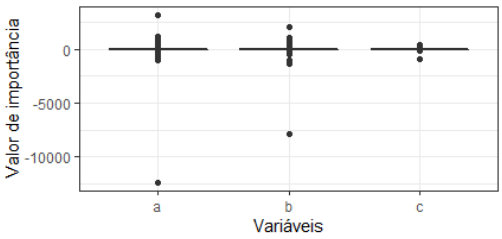
(f) #C4 - Yoon



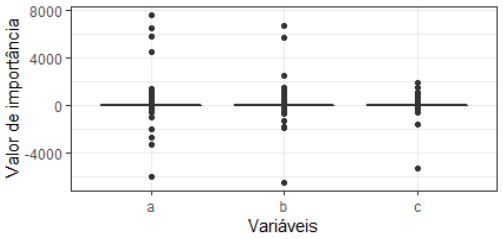
(g) #R4 - Tsaur



(h) #C4 - Tsaur



(i) #R4 - Howes



(j) #C4 - Howes

Figura 4.5: Boxplot dos valores de importância obtidos a partir das 500 RNAs conforme os algoritmos de Garson, Olden, Yoon, Tsaur, e de Howes e Crook para o conjunto de dados simulados #R4 e #C4.

O desempenho dos modelos de regressão foi medido utilizando a normalização da métrica de desempenho *Root Mean Squared Error* (RMSE), a raiz do erro médio quadrático da diferença entre a predição e o valor real, computada conforme descrito a seguir:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4-15}$$

em que y_i é o valor real e $\hat{y}_i$ é o valor predito da i-ésima amostra. O valor de RMSE foi normalizado com base na Equação (4-16), de maneira a transformar o RMSE em um intervalo entre 0 e 1 para facilitar a comparação entre os modelos.

$$NRMSE = \frac{RMSE}{max(\mathbf{y}) - min(\mathbf{y})} \tag{4-16}$$

Para os problemas de classificação utilizamos a métrica de acurácia, que calcula a porcentagem de acertos do modelo com base em uma matriz de confusão que estabelece os valores de verdadeiro positivo (VP), verdadeiro negativo (VN), falso positivo (FP), e falso negativo (FN). A matriz de confusão é apresentada na Tabela 4.5 e o cálculo da acurácia é representado pela Equação (4-17).

Tabela 4.5: Matriz de Confusão.

Predição	**Classe real**	
	Positiva	Negativa
Positiva	VP	FN
Negativa	FP	VN

$$\text{acurácia}(\%) = \frac{VP+VN}{VP+VN+FP+FN} \times 100 \tag{4-17}$$

O desempenho das RNAs de cada conjunto de dados foi satisfatório com valores próximos de zero para os problemas de regressão e valores próximo de 100 para os problemas de classificação (Tabela 4.6).

Tabela 4.6: Desempenho das RNAs por conjunto de dados.

Dados	Métrica	Média	Desvio Padrão	Dados	Métrica	Média	Desvio Padrão
#R1	NRMSE	0,158	0,0001	#C1	Acurácia	99,77	0,19
#R2	NRMSE	0,162	0,0008	#C2	Acurácia	98,45	0,32
#R3	NRMSE	0,163	0,0008	#C3	Acurácia	99,28	0,24
#R4	NRMSE	0,094	0,0005	#C4	Acurácia	98,16	0,44

A Tabela 4.7 apresenta a frequência de uso dos valores de *weight decay* para cada conjunto de dados. Conforme relatado anteriormente, ao treinar cada modelo de RNA todos os valores de *weight decay* foram utilizados e o modelo que obtinha a melhor performance era escolhido, de maneira a utilizar sempre os modelos de RNAs mais precisos. Dessa maneira, a frequência de uso dos *weight decay* varia em cada caso. No próximo capítulo nós investigamos o efeito dos valores de *weight decay* na estabilidade dos WBFS. Para os resultados deste capítulo esse efeito não foi investigado, visto que esses experimentos fogem do escopo dos testes para a validação da abordagem de votação.

Tabela 4.7: Frequência de uso dos valores de *weight decay* nos dados simulados.

Problemas de Regressão	0,1	0,01	0,001	0,0001	0,00001
#R1	99,4%	-	0,6%	-	-
#R2	21,4%	14,4%	1,6%	48%	14,6%
#R3	94,8%	3%	-	1,4%	0,8%
#R4	78%	18,6%	1,4%	0,4%	1,6%
Problemas de Classificação					
#C1	3,8%	15,6%	39,8%	20%	20,8%
#C2	35,2%	21,4%	17,4%	17,4%	8,6%
#C3	29,6%	29%	16,2%	13,4%	11,8%
#C4	5,8%	43,6%	32%	14,2%	4,4%

Resultados da nova abordagem de avaliação dos WBFS

A proposta da abordagem de votação foi aplicada nos rankings de importância dos cinco algoritmos WBFS para todos os conjuntos de dados simulados. O valor de confiança mediu a taxa de votação de uma variável para uma posição específica, levando em consideração 500 RNAs treinadas. As posições de importância mais votadas foram utilizadas para determinar a ordem de importância final.

As Tabelas 4.8 e 4.9 apresentam os resultados da votação para as posições de importância de cada variável, referente aos WBFS de Garson, Olden, Yoon, Tsaur e de Howes para os conjuntos de dados simulados em problemas de regressão e de classificação, respectivamente. As células sombreadas representam a ordem de importância esperada conforme a correlação entre as variáveis independentes e a variável dependente, apresentada anteriormente na Tabela 4.3. Para os algoritmos que não são capazes de identificar a relação negativa entre as variáveis, i.e, o algoritmo de Garson, consideramos a importância absoluta das variáveis. Os valores em negrito representam a maior taxa de votação para a variável em questão.

Observa-se que praticamente todos os rankings de importância gerados pelas 500 RNAs em conjunto com os WBFS de Garson, Olden e Yoon para os conjuntos de dados

com relação linear para os problemas de regressão (#R1 na Tabela 4.8) concordaram que a ordem de importância das variáveis é $a > b > c$. Já para os algoritmos de Tsaur e Howes os resultados da ordem de importância de alguns rankings divergem. O mesmo ocorre para os conjuntos de dados com relação linear para os problemas de classificação (#C1 na Tabela 4.9), com a diferença que alguns resultados das posições de importância dos algoritmos de Garson, Olden e Yoon também divergiram.

Nota-se que as relações negativa quadrática foram identificadas corretamente por um número maior de votos pelos WBFS do que as relações positivas quadráticas para ambos problemas de classificação e regressão (#R2, #R3, #C2 e #C3). Além disso, observa-se que os resultados das votações dos algoritmos de Olden e de Yoon são iguais nas duas tabelas para todos os conjuntos de dados. Isso ocorre devido ao algoritmo de Yoon ser uma normalização do algoritmo de Olden, conforme discutido no Capítulo 2.

Conforme identificado pela correlação de Pearson na Tabela 4.3 ao apresentar os dados simulados, a relação iterativa entre as variáveis a e b possui importância aproximada em relação à variável dependente y e w. Os algoritmos de Garson, Olden e Yoon identificaram corretamente essa aproximação, sendo que Garson atribui as posições de importâncias 1 e 2 apenas para as variáveis a e b (com exceção de apenas uma indicação à posição de importância nº 3 em #R4), enquanto os algoritmos de Olden e Yoon tiveram algumas indicações a mais das variáveis a e b na posição de importância nº 3. Os algoritmos de Tsaur e de Howes apresentaram a estabilidade mais baixa para esses conjuntos de dados, apresentando a quantidade de votos bem distribuída entre as variáveis e posições de importância.

Os algoritmos de Garson, Olden e de Yoon resultaram em uma taxa de votação de uma variável para a mesma posição de importância em todas as 500 RNAs ($\varepsilon = 1$) para a o conjunto de dados #R1 nas três variáveis. A menor taxa média de ε, número de diferentes posições indicadas para a mesma variável, foi de ($1{,}79 \pm 0{,}65$) para o algoritmo de Garson, seguido pelos algoritmos de Olden ($2{,}87 \pm 0{,}82$), Yoon ($2{,}37 \pm 0{,}82$), Howes ($2.6 \pm 0{,}6$), e pelo algoritmo de Tsaur ($2{,}87 \pm 0{,}33$).

A apresentação dos resultados das votações nas Tabelas 4.8 e 4.9 teve como objetivo apresentar ao leitor como a abordagem de votação funciona. Entretanto, avaliar tais tabelas é dispendioso, principalmente se o número de variáveis for grande. Dessa forma, as métricas propostas nessa tese resumem as informações que podem ser obtidas nessas tabelas.

A Tabela 4.10 apresenta os valores de confiança de cada variável em cada conjunto de dados, acompanhada do coeficiente de correlação de *Pearson* identificada por (Corr.) que relacionada a ordem de importância final obtida pela abordagem de votação e a ordem de importância real entre as variáveis.

As variáveis a e b possuem um valor de importância maior do que a variável c nos

Tabela 4.8: Resultado da votação das posições de importância para os conjuntos de dados de problemas de regressão.

		GARSON				OLDEN				YOON				TSAUR				HOWES		
		a	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*
#R1	*1°*	**500**	0	0	*1°*	**500**	0	0	*1°*	**500**	0	0	*1°*	**499**	1	0	*1°*	**492**	0	**8**
	2°	0	**500**	0	*2°*	0	**500**	0	*2°*	0	**500**	0	*2°*	1	**496**	3	*2°*	0	**500**	0
	3°	0	0	**500**	*3°*	0	0	**500**	*3°*	0	0	**500**	*3°*	0	3	**497**	*3°*	**8**	0	**492**
		a	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*
#R2	*1°*	**495**	5	0	*1°*	187	**232**	81	*1°*	187	**232**	81	*1°*	**211**	214	75	*1°*	**492**	6	23
	2°	5	**451**	44	*2°*	**208**	221	71	*2°*	**208**	221	71	*2°*	125	**230**	145	*2°*	8	**450**	42
	3°	0	44	**456**	*3°*	105	47	**348**	*3°*	105	47	**348**	*3°*	164	56	**280**	*3°*	0	44	**456**
		a	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*
#R3	*1°*	**500**	0	0	*1°*	6	**494**	0	*1°*	6	**494**	0	*1°*	46	**454**	0	*1°*	0	11	**489**
	2°	0	**489**	11	*2°*	0	0	**500**	*2°*	0	0	**500**	*2°*	99	41	**360**	*2°*	2	**487**	11
	3°	0	11	**489**	*3°*	**494**	6	0	*3°*	**494**	6	0	*3°*	**355**	5	140	*3°*	**498**	2	0
		a	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*
#R4	*1°*	**452**	48	0	*1°*	**327**	168	5	*1°*	**327**	168	5	*1°*	**217**	264	19	*1°*	25	22	**453**
	2°	47	**451**	2	*2°*	170	**326**	4	*2°*	170	**326**	4	*2°*	77	75	**348**	*2°*	180	**320**	0
	3°	1	1	**498**	*3°*	3	6	**491**	*3°*	3	6	**491**	*3°*	206	**161**	133	*3°*	**295**	158	47

Tabela 4.9: Resultado da votação das posições de importância para os conjuntos de dados de problemas de classificação.

		GARSON				OLDEN				YOON				TSAUR				HOWES		
		a	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*
#C1	*1°*	**500**	0	0	*1°*	**500**	0	0	*1°*	**500**	0	0	*1°*	**278**	114	108	*1°*	100	2	**398**
	2°	0	**494**	4	*2°*	0	**499**	1	*2°*	0	**499**	1	*2°*	160	**253**	87	*2°*	3	**495**	2
	3°	0	4	**496**	*3°*	0	1	**499**	*3°*	0	1	**499**	*3°*	62	133	**305**	*3°*	**397**	3	100
		a	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*
#C2	*1°*	**500**	0	0	*1°*	**373**	121	6	*1°*	**373**	121	6	*1°*	**269**	146	85	*1°*	**405**	1	94
	2°	0	**479**	21	*2°*	94	**376**	30	*2°*	94	**376**	30	*2°*	62	**221**	217	*2°*	0	**494**	6
	3°	0	21	**479**	*3°*	33	3	**464**	*3°*	33	3	**464**	*3°*	169	133	**198**	*3°*	95	5	**400**
		a	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*
#C3	*1°*	**500**	0	0	*1°*	2	**495**	3	*1°*	2	**495**	3	*1°*	205	**230**	65	*1°*	25	6	**469**
	2°	0	**486**	14	*2°*	7	5	**488**	*2°*	7	5	**488**	*2°*	52	188	**260**	*2°*	2	**493**	5
	3°	0	14	**486**	*3°*	**491**	0	9	*3°*	**491**	0	9	*3°*	**243**	82	175	*3°*	**473**	1	26
		a	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*		*a*	*b*	*c*
#C4	*1°*	**266**	234	0	*1°*	**233**	206	61	*1°*	**233**	206	61	*1°*	**173**	**206**	121	*1°*	110	105	**285**
	2°	234	**266**	0	*2°*	214	**225**	61	*2°*	214	**225**	61	*2°*	158	130	**212**	*2°*	**217**	**116**	211
	3°	0	0	**500**	*3°*	53	69	**378**	*3°*	53	69	**378**	*3°*	169	164	167	*3°*	173	**116**	211

conjuntos de dados #R1, #C1, #R2, #C2, #R4 e #C4. Já para os conjuntos de dados #R3 e #C3, a variável a apresenta uma importância negativa, tornando essa variável a terceira mais importante ao considerar a direção de importância (positiva/negativa). Dessa forma, para os algoritmos que conseguem identificar a direção de importância das variáveis, a ordem de importância dos conjuntos de dados #R3 e #C3 utilizada como referência para verificar a correlação entre o ranking de importância final e o ranking de importância real foi $b > c > a$, enquanto que para o algoritmo de Garson, utilizamos como referência a ordem de importância $a > b > c$.

Os maiores valores de confiança foram obtidos pelo algoritmo de Garson (média $\pm$ desvio padrão de (0,94 $\pm$ 0,12), seguido pelos algoritmos de Howes (0,85 $\pm$ 0,17), Olden (0,82 $\pm$ 0,21), Yoon (0,82 $\pm$ 0,21) e Tsaur (0,59 $\pm$ 0,19), ao levar em consideração os resultados obtidos em todos os conjuntos de dados de classificação e de regressão. O valor de confiança mínimo em todos os conjuntos de dados foi de $\beta = 0,33$. Os algoritmos de Garson, Olden, Yoon, e de Howes resultaram em valores bem acima do valor mínimo na maioria dos casos (Figura 4.6), indicando que a maioria das posições de importância obtiveram um alta taxa de votação em apenas uma posição de importância.

O conjunto de dados #C4 foi o que obteve os valores de confiança mais baixos entre os resultados de Garson. As variáveis a e b possuem uma relação interativa, não sendo possível discernir qual delas é mais importante. O algoritmo identificou essas duas variáveis como as duas variáveis principais em todos os modelos treinados, e identificou a variável c como a terceira mais importante. Conforme apresentado na Tabela 4.9, as variáveis a e b obtiveram quantidade de votos aproximadas para as posições de importância 1 e 2. Isso indica que o algoritmo teve alguma dificuldade em determinar qual variável entre a e b era a mais importante.

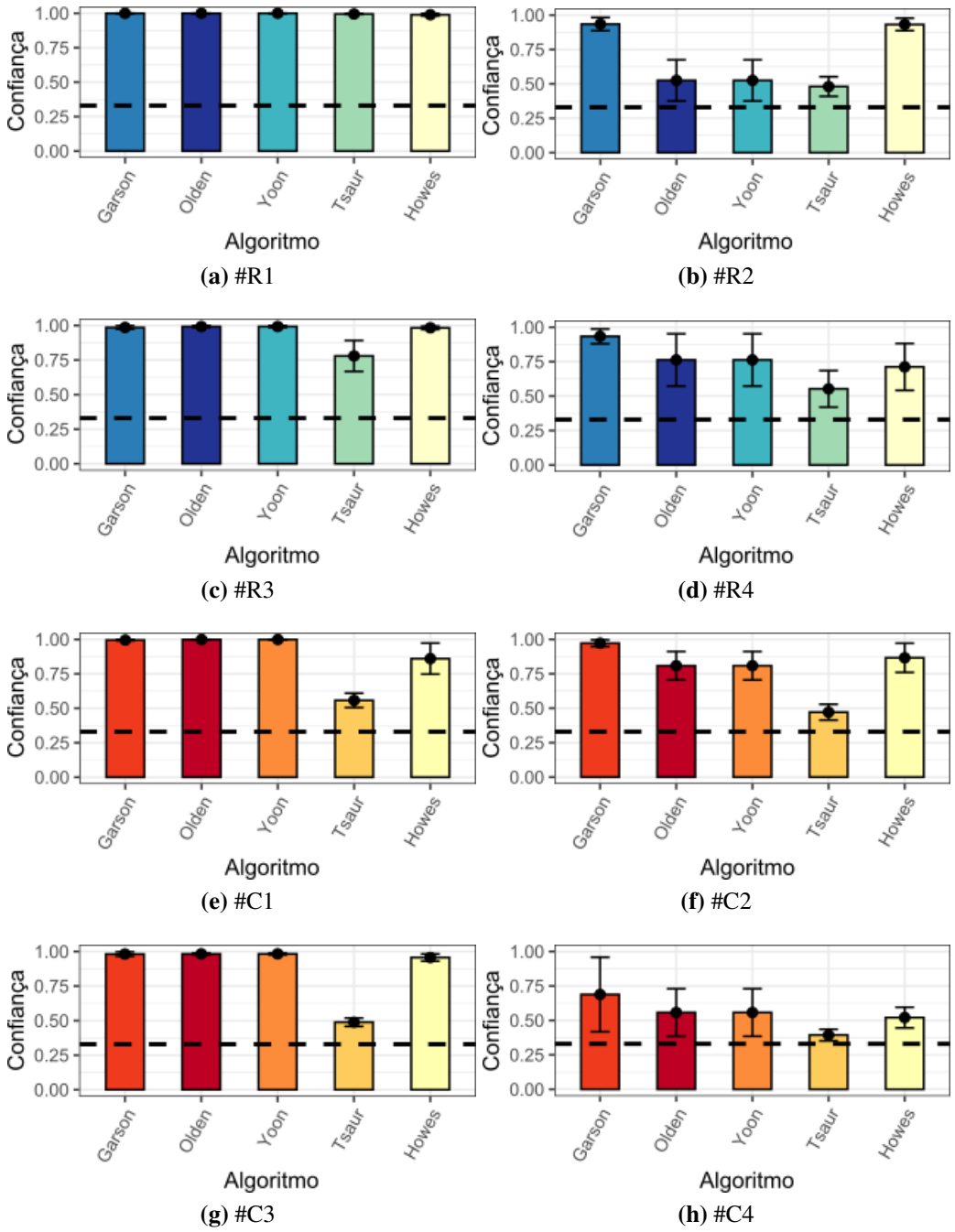
Apesar dessa dificuldade, a abordagem de votação revelou que o algoritmo de Garson identificou a e b como as duas variáveis mais importantes, e identificou a variável c como a terceira mais importante. Os outros algoritmos apontaram as três posições de importância para todas as variáveis ($\varepsilon = 3$ para as três variáveis para os algoritmos de Olden, Yoon, Tsaur e Howes nos conjuntos de dados #R4 e #C4).

Os algoritmos de Olden e Yoon identificaram corretamente a ordem de importância para os problemas de classificação e regressão com base em relações lineares, quadrática negativa e relações interativas. Esses dois algoritmos tiveram resultados semelhantes, o que pode ser explicado devido ao algoritmo de Yoon ser uma normalização do algoritmo de Olden. O algoritmo de Tsaur identificou corretamente os rankings de importância para os conjuntos de dados 1, 2 e 3, para problemas de regressão e classificação. Por último, o algoritmo de Howes apenas identificou a ordem de importância correta para a relação linear para o problema de regressão e para a relação quadrática positiva para ambos os problemas de regressão e classificação.

Tabela 4.10: Valores de confiança e taxas de correlação para os conjuntos de dados simulados de regressão e classificação.

		Problemas de Regressão				Problemas de Classificação			
	WBFS	**a**	**b**	**c**	**Cor.**	**a**	**b**	**c**	**Cor.**
	Garson	1	1	1	1	1	0,992	0,992	1
#R1 e #C1	**Olden**	1	1	1	1	1	0,998	0,998	1
(relação	**Yoon**	1	1	1	1	1	0,998	0,998	1
linear)	**Tsaur**	0,998	0,992	0,994	1	0,556	0,556	0,610	1
	Howes	0,984	1	0,984	1	0,794	0,990	0,796	-1
	Garson	0,990	0,902	0,912	1	1	0,958	0,958	1
#R2 e #C2	**Olden**	0,416	0,464	0,696	0,5	0,746	0,752	0,928	1
(relação pos.	**Yoon**	0,416	0,464	0,696	0,5	0,746	0,752	0,928	1
quadrática)	**Tsaur**	0,422	0,460	0,560	1	0,538	0,442	0,434	0,8
	Howes	0,984	0,900	0,912	1	0,810	0,988	0,8	1
	Garson	1	0,978	0,978	1	1	0,972	0,972	1
#R3 e #C3	**Olden**	0,988	0,988	1	1	0,982	0,990	0,967	1
(relação neg.	**Yoon**	0,988	0,988	1	1	0,982	0,990	0,967	1
quadrática)	**Tsaur**	0,710	0,908	0,720	1	0,486	0,460	0,520	1
	Howes	0,996	0,974	0,978	0,5	0,946	0,986	0,938	0,5
	Garson	0,904	0,902	0,996	1	0,532	0,532	1	1
#R4 e #C4	**Olden**	0,654	0,652	0,982	1	0,466	0,450	0,756	1
(relação	**Yoon**	0,654	0,652	0,982	1	0,466	0,450	0,756	1
iterativa)	**Tsaur**	0,434	0,528	0,696	0,8	0,346	0,412	0,424	0,8
	Howes	0,509	0,640	0,906	-0,5	0,434	0,558	0,570	-0,8

A abordagem de utilizar a média dos valores de importância obtidos por um WBFS aplicado a um conjunto de RNAs foi realizada nas mesmas 500 RNAs utilizadas na abordagem de votação (Categoria C de uso dos WBFS apresentada no Capítulo 3) para comparar os resultados com os da metodologia proposta. Os resultados dos valores de importância média de Garson indicaram corretamente a ordem $a > b > c$ para todos os conjuntos de dados (Figura 4.7). Os resultados dos valores de importância médios de



(a) #R1 **(b)** #R2

(c) #R3 **(d)** #R4

(e) #C1 **(f)** #C2

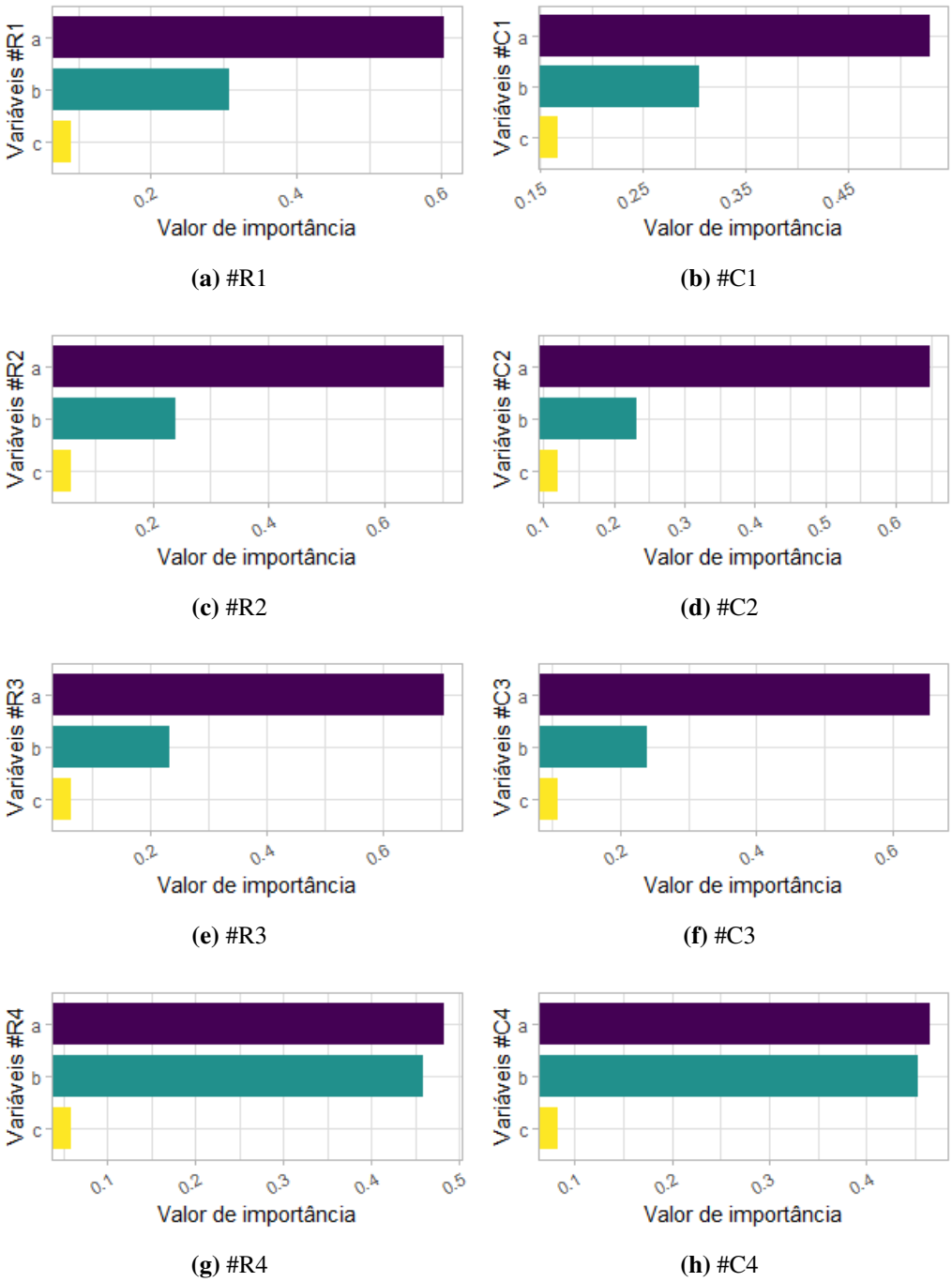
(g) #C3 **(h)** #C4

Figura 4.6: Comparação dos resultados do valor de confiança dos cinco algoritmos WBFS. A linha tracejada representa a confiança mínima.

Olden indicaram corretamente a ordem de importância para todos os conjuntos de dados (Figura 4.8), exceto para #R2, #C2 e #C4. No entanto, o resultado desse algoritmo na abordagem do valor médio superestimou incorretamente a importância de a para a relação quadrática positiva e para a relação interativa no problema de classificação.

Os valores médios de importância de Yoon indicaram corretamente a ordem das variáveis para quase todos os conjuntos de dados (Figura 4.9). A exceção ocorreu no

problema de regressão com a relação quadrática positiva (conjunto de dados #R2), em que a ordem de importância esperada era $a > b > c$ e a ordem de importância resultante foi $b > c > a$. Os demais algoritmos identificaram incorretamente a magnitude e a relação positiva/negativa entre as variáveis na maioria dos casos (Figuras 4.10 e 4.11).



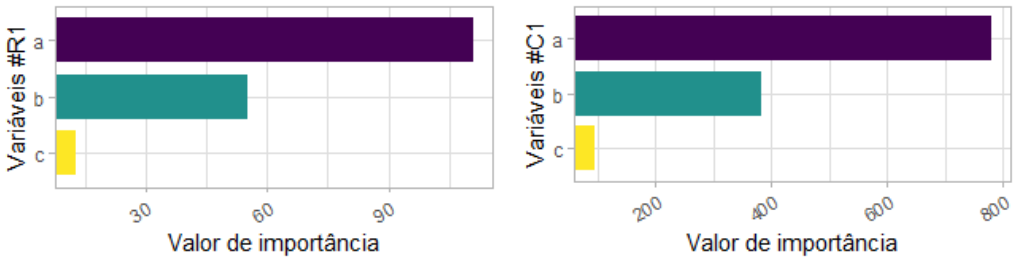
(a) #R1 **(b)** #C1

(c) #R2 **(d)** #C2

(e) #R3 **(f)** #C3

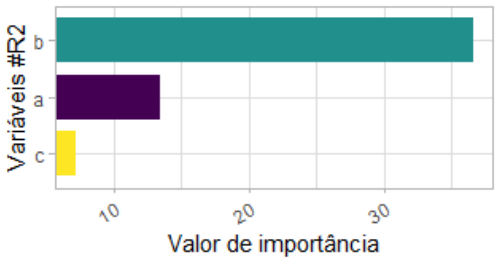
(g) #R4 **(h)** #C4

Figura 4.7: Valores médios de importância de Garson para os conjuntos de dados simulados

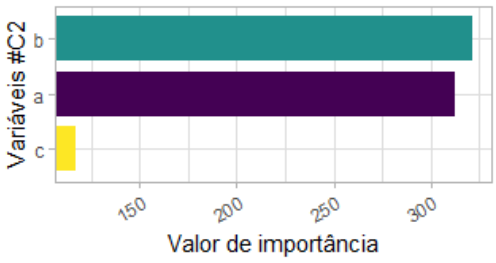


(a) #R1

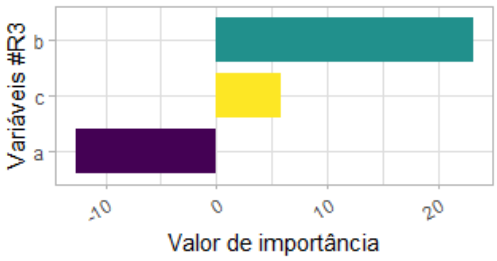
(b) #C1



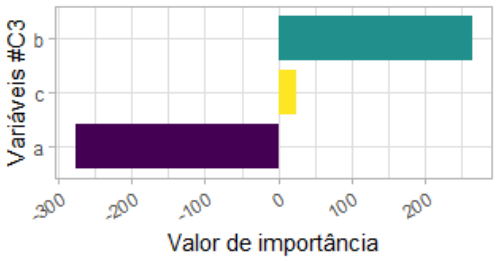
(c) #R2



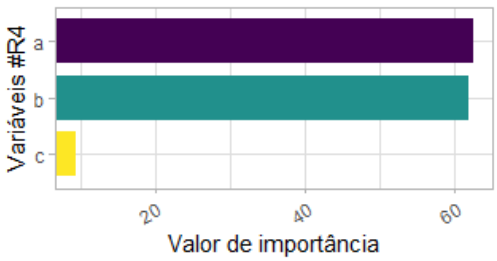
(d) #C2



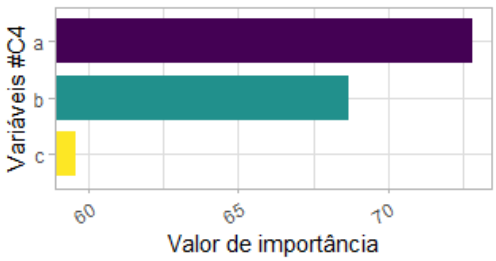
(e) #R3



(f) #C3

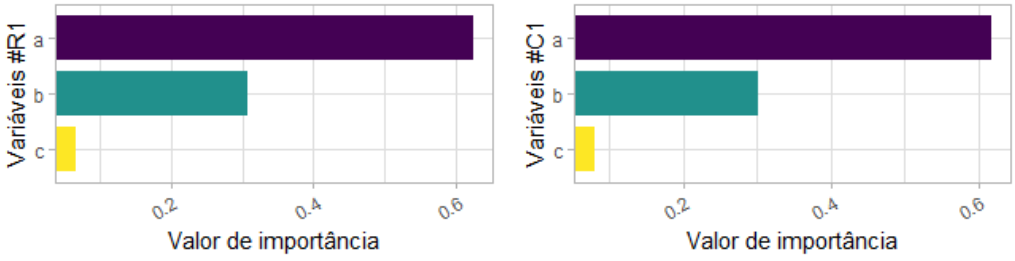


(g) #R4



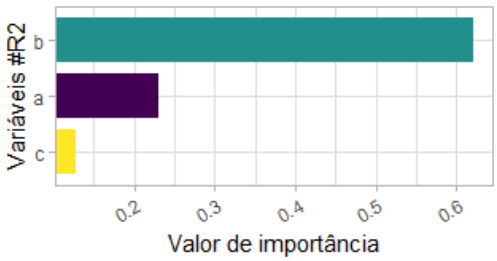
(h) #C4

Figura 4.8: Valores médios de importância de Olden para os conjuntos de dados simulados

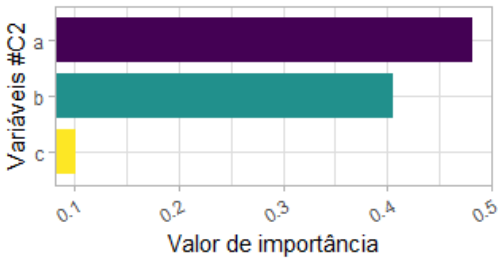


(a) #R1

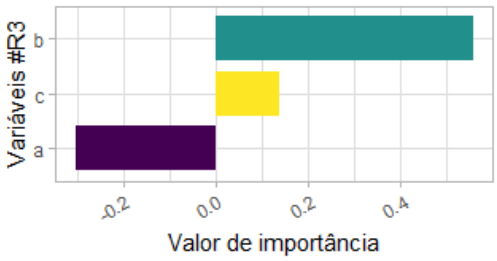
(b) #C1



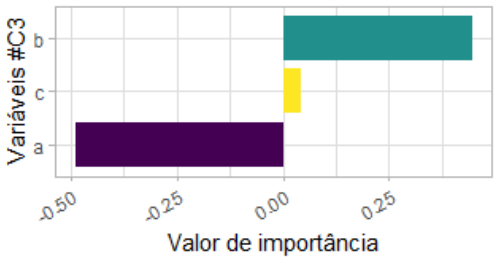
(c) #R2



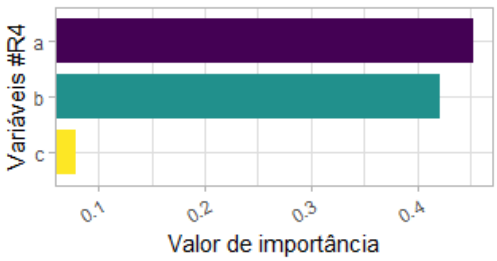
(d) #C2



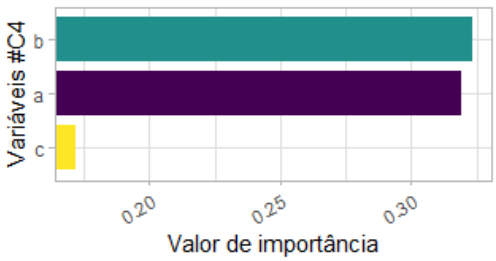
(e) #R3



(f) #C3

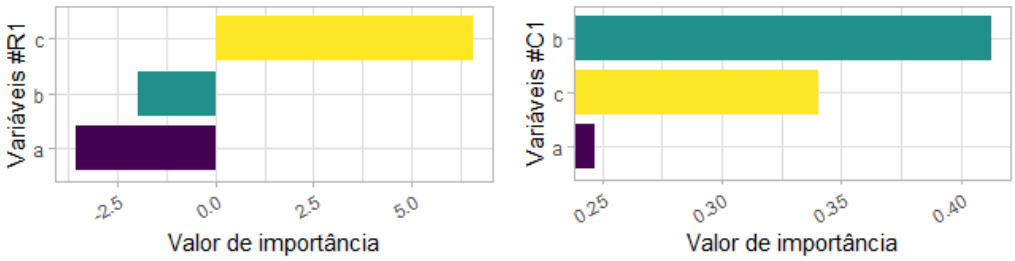


(g) #R4



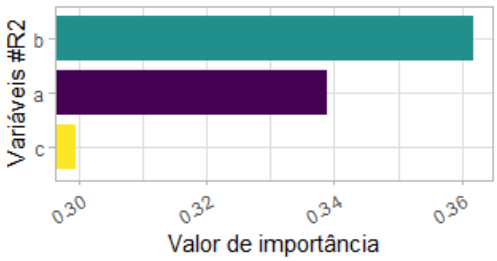
(h) #C4

Figura 4.9: Valores médios de importância de Yoon para os conjuntos de dados simulados

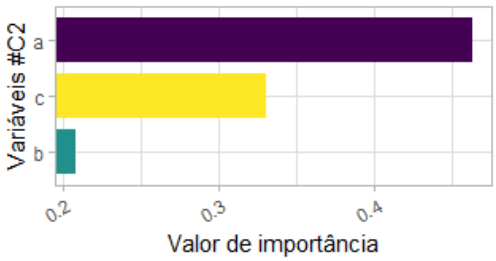


(a) #R1

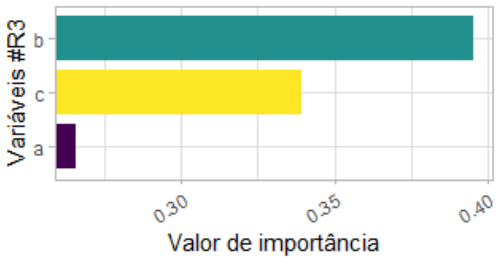
(b) #C1



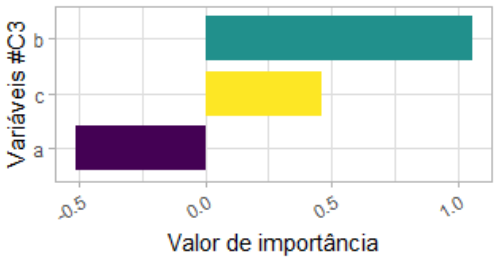
(c) #R2



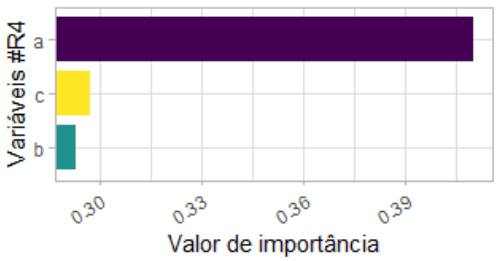
(d) #C2



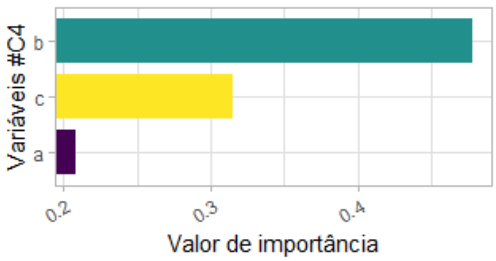
(e) #R3



(f) #C3

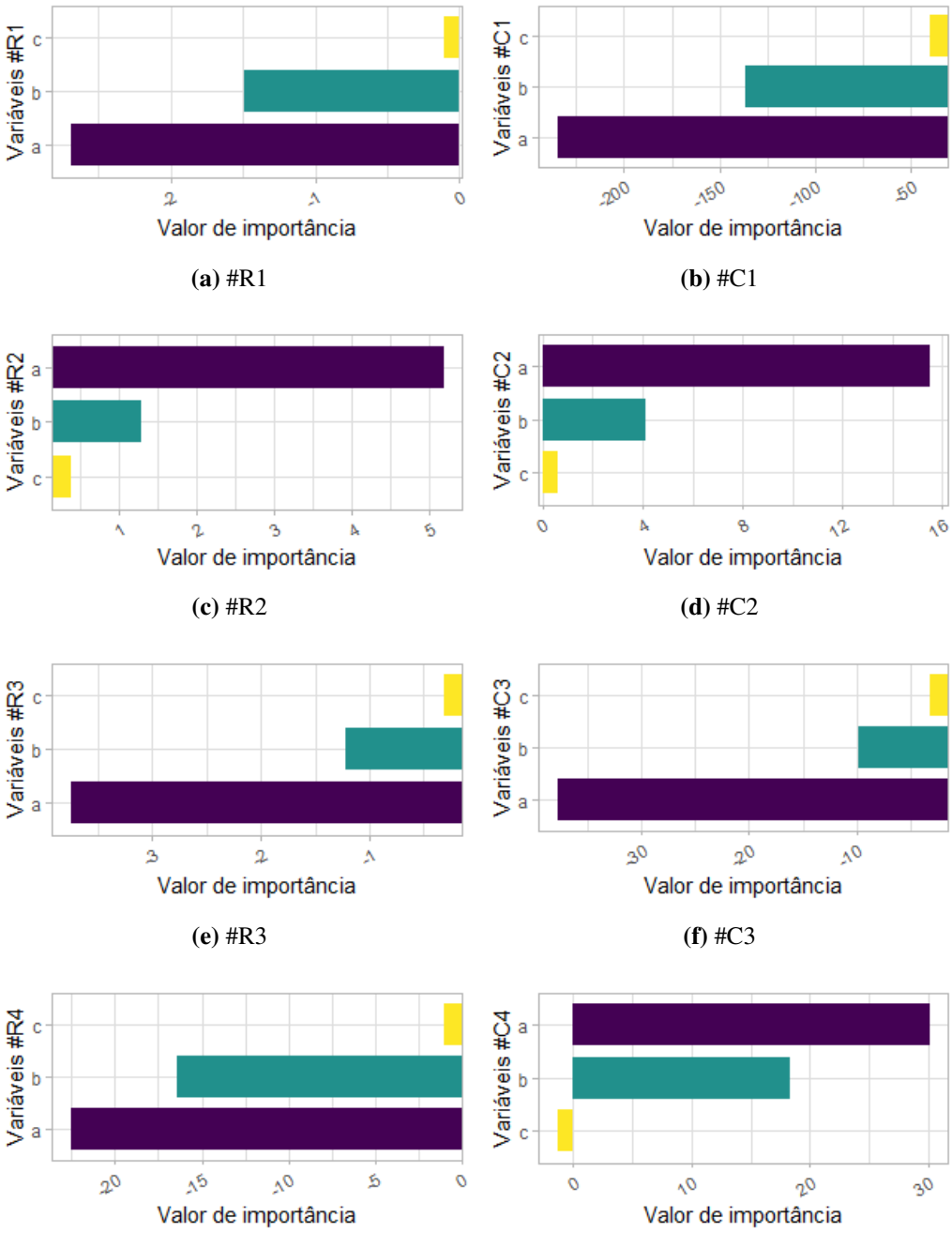


(g) #R4



(h) #C4

Figura 4.10: Valores médios de importância de Tsaur para os conjuntos de dados simulados



(a) #R1 **(b)** #C1

(c) #R2 **(d)** #C2

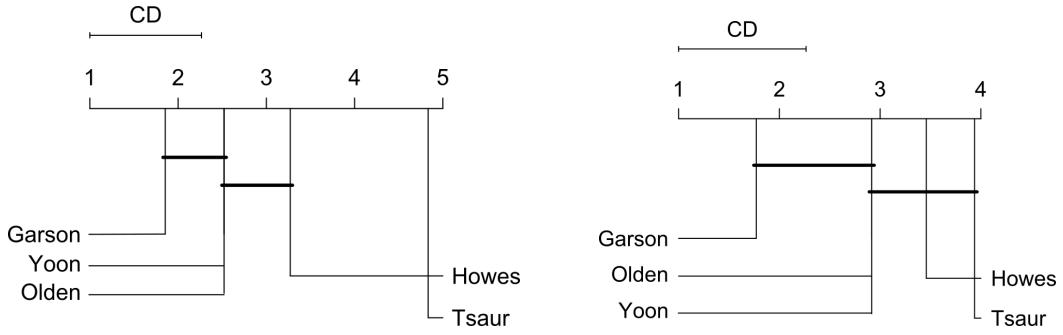
(e) #R3 **(f)** #C3

(g) #R4 **(h)** #C4

Figura 4.11: Valores médios de importância de Howes e Crook para os conjuntos de dados simulados

Em seguida, nós aplicamos o teste de Friedman para verificar a hipótese nula de que todos os algoritmos possuem valores de confiança e valores ε equivalentes. O teste resultou em um valor de p significativo para ambos valores de confiança (valor $p < 0{,}001$, $\chi^2 = 55{,}02$) e valores ε (valor $p < 0.001$, $\chi^2 = 44.27$), indicando que existem pelo menos dois algoritmos que tiveram desempenhos diferentes. O teste *post-hoc* de Nemenyi demonstrou que os algoritmos de Garson, Olden e de Yoon apresentam

resultados equivalentes do ponto de vista estatístico, enquanto que esses algoritmos demonstraram-se estatisticamente diferentes do algoritmos de Tsaur, e apenas o algoritmo de Garson possui diferença estatística em relação ao algoritmo de Howes (Figura 4.12a). Este resultado indica que os algoritmos de Garson, Olden e de Yoon forneceram valores de confiança mais elevados para os conjuntos de dados simulados. Os resultados foram semelhantes em relação ao número de diferentes posições indicadas para uma mesma variável (ε). O teste de Nemenyi revelou que o algoritmo de Garson obteve o menor ranking médio, sendo este equivalente aos algoritmos de Olden e de Yoon, com uma diferença estatística significativa apenas em relação aos algoritmos de Howes e de Tsaur (Figura 4.12b).



(a) Resultado sobre os valores de confiança **(b)** Resultado sobre os valores ε

Figura 4.12: Representação gráfica dos resultados do teste *post-hoc* de Nemenyi. Métodos que não são significativamente diferentes (no nível $p < 0{,}05$) são conectados por uma barra.

4.4.2 Experimentos numéricos com dados do UCI

Com relação aos resultados das bases de dados simulados, utilizamos apenas os algoritmos de Garson, Olden e de Yoon para avaliar as bases de dados do repositório UCI considerando que esses três algoritmos resultaram nos maiores valores de confiança.

Os valores de confiança obtidos nas bases de dados do repositório UCI foram inferiores aos valores de confiança obtidos nas bases de dados simulados (Figura 4.13). A figura mostra os valores de confiança para cada variável de cada banco de dados. A linha tracejada indica o valor mínimo de confiança para o conjunto de dados. A maioria dos valores de confiança, especialmente para as bases de dados Parkinson e Wine Quality, foram os mais próximos do valor de confiança mínimo.

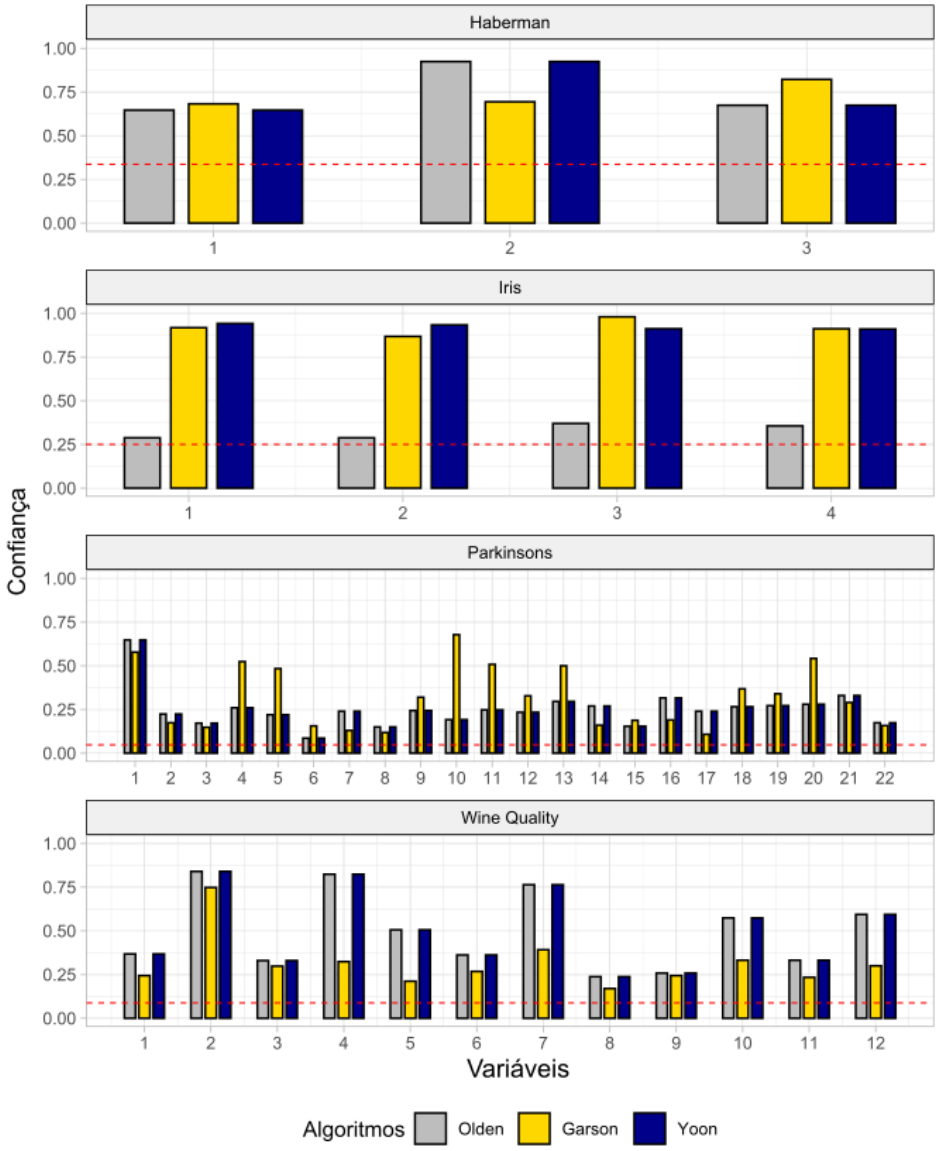


Figura 4.13: Valores de confiança por variável para os conjuntos de dados do repositório UCI.

Os valores ε de cada variável foram altos em todos os conjuntos de dados (Figura 4.14). O valor máximo de ε para os conjuntos de dados Parkinson, Haberman, Iris e Wine Quality foram 22, 3, 4 e 12, respectivamente. Essa métrica atingiu o valor máximo em todas as variáveis para o conjunto de dados Haberman para os três algoritmos (Tabela 4.11). Os valores ε resultantes dos algoritmos de Olden e Yoon também alcançaram o valor máximo no conjunto de dados Iris. Esses resultados indicam que todas as variáveis tiveram votos em todas, ou quase todas, posições de importância. Os valores de ε para os conjuntos de dados Parkinson e Wine Quality também foram os mais próximos dos valores máximos, o que indica a instabilidade dos algoritmos estudados por uma alta taxa de posições de importância votadas para todas as variáveis.

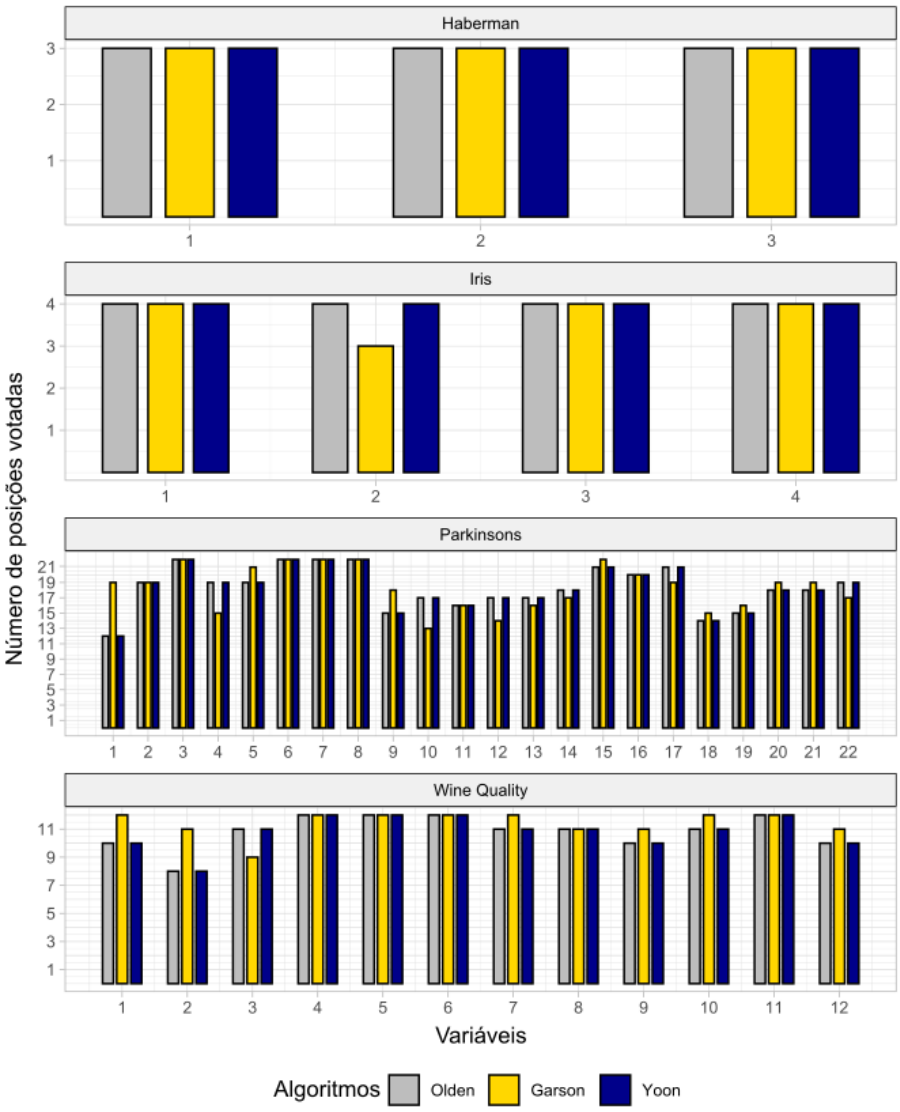


Figura 4.14: Valores ε por variável para as bases de dados Haberman, Iris, Parkinson, e Wine Quality.

Tabela 4.11: Resultados obtidos para os dados do UCI *Machine Learning Repository*.

Banco de dados	Garson	Olden	Yoon
Parkinson			
Confiança (média ± d.p.)	0,317 ± 0,178	0,25±0,106	0,25±0,106
Valores ε (média±d.p.)	18,31±2,8	18,31±2,8	18,31±2,8
Haberman			
Confiança (média±d.p.)	0,732±0,077	0,748±0,153	0,748±0,153
Valores ε (média±d.p.)	3±0	3±0	3±0
Iris			
Confiança (média±d.p.)	0,919±0,046	0,325±0,043	0,924±0,046
Valores ε (média±d.p.)	3,75±0,5	4±0	4±0
Wine Quality			
Confiança (média±d.p.)	0,313±0,15	0,499±0,218	0,25±0,106
Valores ε (média±d.p.)	11,41±0,90	10,83±1,19	18,31±2,8

O teste de Friedman foi aplicado nos resultados de confiança e valores ε para os bancos de dados do repositório UCI. Os resultados do teste para os valores de confiança (valor p = 0,2367, χ^2 = 2,8819) e para os valores ε (valor p = 0,3372, χ^2 = 2,1739) indicaram que do ponto de vista estatístico, a hipótese nula de que os algoritmos de Garson, Olden e Yoon não apresentaram diferença significativa não deve ser rejeitada. Ou seja, os algoritmos de Garson, Olden e Yoon foram equivalentes nos conjuntos de dados testados.

A abordagem dos valores médios de importância também foi aplicada para os conjuntos de dados UCI (Tabelas 4.12, 4.13 e 4.14). Os resultados indicaram que 10 de 22 variáveis resultaram na mesma posição de importância no rankings da abordagem de valor de importância médio e na abordagem de votação para o conjunto de dados Parkinson para o algoritmo de Garson (Tabela 4.12); 5 de 22 variáveis para o algoritmo de Olden (Tabela 4.13); e 3 de 22 variáveis para o algoritmo de Yoon (Tabela 4.14).

A mesma ordem de importância foi obtida em ambas as abordagens para o conjunto de dados Haberman para todos os algoritmos. Uma ordem de importância semelhante foi obtida em ambas as abordagens do conjunto de dados Iris para algoritmos de Garson e Yoon (Tabelas 4.12 e 4.14). O algoritmo de Olden resultou em ordens de importância completamente diferentes para este conjunto de dados (Tabela 4.13, conjunto de dados Iris).

Por último, 9 de 12 variáveis da base Wine Quality tiveram a mesma posição de importância para ambas as abordagens do algoritmo de Garson (Tabela 4.12); 1 de 12 variáveis para o algoritmo de Olden (Tabela 4.13); e 10 de 12 para o algoritmo de Yoon (Tabela 4.14).

Mesmo que não tenha sido identificado nenhuma diferença significativa entre os valores de confiança dos algoritmos de Garson, Olden e Yoon nos conjuntos de dados estudados, esses algoritmos resultaram em rankings de importância diferentes, para as quais as variáveis não estavam nas mesmas posições de importância.

Tabela 4.12: Posição de importância média para as bases de dados UCI de acordo com o algoritmo de Garson. As variáveis em destaque foram aqueles que tiveram a mesma posição de importância na abordagem de votação.

Ranking	Parkinson	Haberman	Iris	Wine Quality
1	D2	Nodes	Petal length	chlorides
2	HNR	Age	Petal width	density
3	spread1	Year	Sepal width	Volatile acidity
4	RPDE		Sepal length	sulphates
5	DFA			pH
6	MDVP Shimmer dB.			alcohol
7	spread2			Fixed acidity
8	PPE			quality
9	MDVP Flo Hz			Free sulfur dioxide
10	MDVP Fo Hz			Citric acid
11	MDVP Fhi Hz			Residual sugar
12	Shimmer DDA			Total sulfur dioxide
13	MDVP Shimmer			
14	MDVP APQ			
15	Shimmer APQ5			
16	NHR			
17	Shimmer APQ3			
18	Jitter DDP			
19	MDVP Jitter			
20	MDVP RAP			
21	MDVP PPQ			
22	MDVP Jitter Abs			

Tabela 4.13: Posição de importância média para as bases de dados UCI de acordo com o algoritmo de Olden. As variáveis em destaque foram aqueles que tiveram a mesma posição de importância na abordagem de votação.

Ranking	Parkinson	Haberman	Iris	Wine Quality
1	D2	Nodes	Sepal width	density
2	spread1	Age	Sepal length	alcohol
3	MDVP Shimmer dB	Year	Petal width	Citric acid
4	DFA		Petal length	Residual sugar
5	spread2			Total sulfur dioxide
6	PPE			Free sulfur dioxide
7	RPDE			quality
8	Shimmer DDA			Fixed acidity
9	MDVP Shimmer			pH
10	MDVP.APQ			sulphates
11	Shimmer APQ5			Volatile acidity
12	Shimmer APQ3			chlorides
13	Jitter DDP			
14	MDVP RAP			
15	MDVP PPQ			
16	MDVP Jitter			
17	MDVP Jitter Abs			
18	MDVP Fhi Hz			
19	MDVP Fo Hz			
20	NHR			
21	HNR			
22	MDVP Flo Hz			

Tabela 4.14: Posição de importância média para as bases de dados UCI de acordo com o algoritmo de Yoon. As variáveis em destaque foram aqueles que tiveram a mesma posição de importância na abordagem de votação.

Ranking	Parkinsons	Haberman	Iris	Wine Quality
1	D2	Nodes	Sepalwidth	density
2	spread1	Age	Sepal length	alcohol
3	MDVP Shimmer dB	Year	Petal width	Citric acid
4	DFA		Petal length	Residual sugar
5	spread2			Total sulfur dioxide
6	PPE			Free sulfur dioxide
7	RPDE			quality
8	Shimmer DDA			Fixed acidity
9	MDVP Shimmer			pH
10	MDVP APQ			sulphates
11	Shimmer APQ5			Volatile acidity
12	Shimmer APQ3			chlorides
13	Jitter DDP			
14	MDVP RAP			
15	MDVP PPQ			
16	MDVP Jitter			
17	MDVP Jitter Abs			
18	MDVP Fhi Hz			
19	MDVP Fo Hz			
20	NHR			
21	HNR			
22	MDVP Flo Hz			

4.5 Discussão

Os resultados dos experimentos realizados neste capítulo confirmam a alta variabilidade dos valores de importância obtidos do WBFS, conforme demonstrado por outros estudos anteriores [97, 27]. A nova metodologia proposta demonstrou-se capaz de medir a estabilidade de métodos WBFS por meio de uma abordagem de votação, juntamente com as novas métricas para medir o desempenho dos algoritmos. A magnitude da importância da variável pode ser enganosa e pode indicar uma ordem de importância errônea devido à diferença entre os pesos da RNA obtidos de diferentes modelos de treinamento. A abordagem de votação, juntamente com a métrica de confiança, revela que

os algoritmos de Garson, Olden e Yoon são os WBFS mais estáveis entre os estudados considerando os conjuntos de dados simulados.

Diante de um conjunto de 500 RNAs treinadas dos dados simuladores, o algoritmo de Garson teve uma confiança média de 0,94, o que mostrou que uma média de 470 rankings de importância indicava a mesma ordem de importância. Os algoritmos de Olden e Yoon resultaram em um número menor de rankings de importância que indicavam a mesma ordem de importância (média de confiança de 0,82, equivalente a aproximadamente 410 rankings de importância para a mesma posição posição de importância para uma dada variável). Os algoritmos de Tsaur resultaram os valores de confiança mais baixos, em que entre todas os rankings de importância com uma média de apenas 295 rankings de importância para a mesma posição de importância.

Os testes estatísticos revelaram que os algoritmos de Garson, Olden e Yoon apresentam diferenças estatísticas não significativas, sendo estes os WBFS que resultaram em valores de confiança mais elevados, considerados equivalentes. O algoritmo de Garson foi o que obteve o ranking médio mais baixo no teste *post-hoc* de Nemenyi; no entanto, a diferença entre Olden e Yoon não atingiu a diferença crítica para designar o algoritmo de Garson como tendo um nível mais alto de estabilidade (Figura 4.12).

O algoritmo de Olden é comumente escolhido por alguns autores em relação ao algoritmo de Garson devido à sua capacidade de identificar a importância negativa das variáveis [39, 93, 106]. No entanto, os resultados mostraram que o algoritmo de Olden identificou incorretamente uma importância negativa no conjunto de dados simulado em algumas variáveis ao usar a abordagem de valores de importância média.

A comparação entre a abordagem de votação e a abordagem dos valores médios de importância nos conjuntos de dados simulados revelam que, apesar dos resultados precisos do algoritmo de Garson, a magnitude da importância pode ser enganosa. Os resultados estão de acordo com o estudo de Fischer [35], que utilizou o valor médio de importância e obteve os mesmos resultados, em que o algoritmo de Garson teve um desempenho melhor do que Olden em todos os casos estudados.

De maneira complementar ao estudo de Fischer [35] que utilizou apenas dados simulados, neste capítulo foi possível identificar uma maior instabilidade dos WBFS ao enfrentar conjuntos de dados do mundo real. Os valores ε mostraram que as variáveis dos conjuntos de dados UCI foram classificadas em praticamente todas as posições de importância possíveis. Isso indica que os rankings de importância gerados por meio de um grande conjunto de RNAs resultaram em classificações diferentes, em que uma variável foi apontada para aproximadamente todas as posições de importância. Isso significa que apenas uma RNA treinada pode resultar em ranking de importância errônea e os valores médios de importância também podem levar a conclusões enganosas. Os resultados demonstraram uma evidência consistente da necessidade de treinar um grande conjunto

de RNAs para avaliar os métodos WBFS.

A grande vantagem de usar a abordagem de votação, juntamente com a confiança e os valores ε, é avaliar a estabilidade e confiabilidade de um ranking de importância gerado por um WBFS. O uso comum de utilizar apenas uma RNA para calcular um ranking a partir de um WBFS, ou o uso dos valores médios de importância pode ser usado por meio de uma seleção de variável do tipo *wrapper*. Dessa maneira, o subconjunto de variáveis estará sujeito à validação de um classificador conforme sua capacidade de predição.

Essa prática é comum e utilizada com sucesso em diversas aplicações relacionadas à ciência de alimentos [7, 129, 17, 23, 18, 24, 19] . Nestes estudos, os quais a maioria foram conduzidos por nós [17, 23, 18, 24, 19], um seletor de variáveis que resulta em um ranking de importância foi utilizado, e, a partir desse ranking diferentes subconjuntos de variáveis foram formados para serem validados por um classificador. Dessa maneira, o subconjunto de variáveis com o maior poder de predição era escolhido.

Essa abordagem pode ser usada em WBFS sem considerar a estabilidade dos métodos, uma vez que o subconjunto de variáveis final é escolhido com base na taxa de predição. No entanto, existem vários autores que usaram o ranking de importância para estabelecer premissas e conclusões sobre a importância das variáveis com base exclusivamente em uma única RNA treinada [4, 87, 90, 93, 145, 148, 155], ou com base nos valores médios de importância [27, 41, 97, 102].

Conforme demonstrado neste estudo, o treinamento de várias RNAs leva a diferentes rankings de importância. Desta forma, as premissas estabelecidas com base em apenas um ranking de importância podem ser diferentes dependendo da inicialização do peso e do processo de treinamento e, consequentemente, não são confiáveis. As premissas estabelecidas com base nos valores médios de importância também podem ser enganosas, pois cada ranking de importância pode apontar uma variável para todas as posições de importância. Consequentemente, esta abordagem não garante a validade dos valores de importância das variáveis.

Existem alguns estudos que compararam os rankings de importância do WBFS com a análise de sensitividade, o perfil de Lek e os métodos de derivadas parciais na tentativa de evitar conclusões enganosas por meio dos valores médios de importância, utilizando a concordância entre esses métodos para estabelecer conclusões sobre os dados [27, 41, 97]. No entanto, esses métodos de análise de sensitividade também demonstraram variabilidade dos valores de importância, e nenhum estudo foi realizado para investigar se todos os rankings apontavam para posições de importância semelhantes.

O algoritmo de Garson foi o algoritmo com os maiores valores de confiança para os conjuntos de dados simulados, seguido pelos algoritmos de Olden e Yoon, que demonstraram um desempenho equivalente do ponto de vista estatístico. No entanto,

ao enfrentar conjuntos de dados do mundo real, o valor de confiança foi menor, muito próximo do valor de confiança mínimo, especialmente ao confrontar um conjunto de dados com um grande número de variáveis. Este resultado indica que ao enfrentar conjuntos de dados mais complexos, o algoritmo de Garson é mais instável do que ao enfrentar conjuntos de dados com relações mais simples. O uso da abordagem de votação, juntamente com o valor de confiança, pode revelar a certeza da ordem de importância apontada por um WBFS.

Dessa maneira, um pesquisador deve avaliar os resultados obtidos por nossa metodologia ao utilizar um WBFS para concluir se o ranking de importância é confiável ou não; se este deve ser utilizado como ferramenta de inferência quanto à importância das variáveis para o problema estudado, ou conforme apontamos, utilizar o ranking de importância como etapa de pré-processamento para geração de um subconjunto de variáveis, a fim de atingir um subconjunto que demonstre maior capacidade preditiva.

4.6 Considerações finais

Esse capítulo apresentou a proposta da abordagem de votação para a avaliação do resultado de um WBFS. Os resultados demonstraram que a abordagem é útil para avaliar a estabilidade dos rankings de importância gerados. Até onde sabemos, esta é a primeira vez que a estabilidade de WBFS foi medida.

A abordagem proposta revela que uma dada variável pode ser classificada em diferentes posições de importância dependendo da RNA utilizada como base. Nesse sentido, o uso do valor médio de importância pode não ser adequado. Além disso, os resultados experimentais mostraram que os algoritmos de Garson, Olden e de Yoon geram rankings de importância com estabilidade semelhante, mas resultam em diferentes posições de importância finais.

Apesar dos resultados positivos em relação à usabilidade da abordagem proposta, o problema da instabilidade dos WBFS não foi resolvido. Esse problema se tornou evidente a partir de nossa abordagem.

Algumas das limitações do estudo realizado neste capítulo é o uso de apenas quatro conjuntos de dados reais, e o fato de que nos baseamos em modelos de RNAs com a melhor capacidade de predição obtidas pela otimização do parâmetro de *weight decay*. Outra limitação dos experimentos deste capítulo é que identificamos que os rankings de importância finais dos WBFS são diferentes, porém, não realizamos nenhum experimento até o momento para avaliar a capacidade de predição dos rankings de importância gerados por esses algoritmos.

Mediante essas limitações, o próximo capítulo foi escrito para superar a limitação do uso de apenas uma arquitetura de RNA. O Capítulo 5 apresenta um estudo detalhado

sobre a influência de diferentes arquiteturas de RNAs na estabilidade dos WBFS. Além disso, também apresentamos uma proposta de melhoria dos algoritmos de Garson, Olden e de Yoon para aumentar sua estabilidade ao lidar com bases de dados reais.

CAPÍTULO 5

Influência de parâmetros da RNA na estabilidade dos WBFS e melhoria dos algoritmos

Neste capítulo investigamos a influência de alguns parâmetros de treinamento de uma RNA na estabilidade de um método WBFS. Sabe-se que a otimização dos parâmetros do treinamento de uma RNA, como o número de neurônios na camada oculta, o número de camada ocultas, e a otimização dos parâmetros de algoritmos que ajustam os pesos de uma RNA podem resultar em um melhor desempenho da rede. No entanto, até onde sabemos não existe nenhum estudo que investiga a relação desses parâmetros com a estabilidade de métodos WBFS. Além disso, neste capítulo também propomos uma melhoria nos algoritmos de Garson, Olden e de Yoon.

5.1 Motivação

Com base no capítulo anterior percebe-se a necessidade de um método de seleção de variáveis baseados nos pesos de conexão de uma RNA que seja estável. Conforme demonstrado em alguns estudos [21, 119], a aplicação dos WBFS em uma única RNA pode gerar resultados errôneos devido à variação que ocorre no ranking de importância gerado por diferentes matrizes de pesos. Dessa maneira, é necessário treinar diversas RNAs para então possibilitar um resultado mais estável.

O algoritmo de Garson demonstrou ser o mais estável dos algoritmos. No entanto, o treinamento de várias RNAs pode levar a alta complexidade computacional a depender do conjunto de dados, tornando inviável a aplicação de Garson e os demais métodos em cenários de dados com alta dimensionalidade.

Após estabelecer a nova metodologia para avaliar a estabilidade dos WBFS, este capítulo investiga alguns aspectos do treinamento da RNA em relação à estabilidade dos métodos. A hipóteses levantadas se baseiam no fato de que a otimização dos parâmetros de treinamento de uma RNA pode levar a um melhor desempenho do modelo. No entanto,

não há evidências de que estes parâmetros também influenciam na estabilidade de um WBFS.

Os aspectos investigados neste capítulo podem ser contextualizadas com as seguintes perguntas:

- Os parâmetros de uma RNA como o número de neurônios ocultos, função de ativação e valor de *weight decay* influenciam na estabilidade de um WBFS?
- O tipo de RNA influencia a estabilidade de um WBFS?
- Qual o número de RNAs devem ser treinadas para que um WBFS atinja sua possível estabilidade em um conjunto de dados?
- Associar um WBFS com outra métrica de importância das variáveis aumenta a estabilidade dos algoritmos?

As respostas a essas perguntas podem levar a melhores práticas de utilização de um método WBFS. Se algum dos parâmetros investigados se demonstrar significativamente efetivo para melhorar a estabilidade de um método WBFS, o pesquisador que utilizar tais métodos pode obter resultados mais confiáveis e possivelmente, requerer um número menor de RNAs para atingir uma estabilidade aceitável.

Outro ponto a ser investigado neste capítulo é a estimação de um número aproximado de RNAs que são necessárias para atingir a estabilidade possível em um dado banco de dados. Esse aspecto é importante pois é inviável treinar um grande número de RNAs em um banco de dados com alta dimensionalidade devido ao custo computacional envolvido.

Este capítulo também apresenta uma proposta de melhoria dos algoritmos de Garson, Olden e de Yoon. A proposta visa melhorar a estabilidade do WBFS ponderando a medida de importância dos algoritmos originais, usando os valores de importância gerados pelo algoritmo ReliefF. Este algoritmo foi escolhido devido à sua simplicidade e eficácia [103]. Além disso, este algoritmo tem sido usado com sucesso em diferentes campos de pesquisa como bioinformática [140], ciência de alimentos [19, 151] e avaliação de risco de crédito [8].

Até onde sabemos, o potencial e as implicações de diferentes modelos de RNA e seus parâmetros de treinamento nos resultados de WBFS ainda não foram estudados. Neste capítulo, abordamos este problema e avaliamos sistematicamente o comportamento de estabilidade de seis algoritmos WBFS (algoritmos de Garson, Olden e Yoon, e as três melhorias propostas). Comparamos a estabilidade dos WBFS em dois modelos diferentes de RNA (*Multilayer Perceptron* - MLP e *Extreme Learning Machines* - ELM) com diferentes cenários de parâmetros de treinamento usando 16 bases de dados do UCI *Machine Learning Repository*.

O algoritmo MLP foi escolhido por ser um dos modelos de RNA *feed-forward* mais comuns, junto com o algoritmo de *backpropagation*, que pode consumir uma enorme

quantidade de tempo computacional devido aos pesos da RNA e otimização de parâmetros [117]. Em contraste, ELM é um modelo de RNA que não requer tal tempo computacional porque seus pesos de entrada são atribuídos aleatoriamente, enquanto apenas os pesos de saída que conectam os nós ocultos e os nós de saída são atualizados [59]. Esse algoritmo tem sido usado com sucesso em muitos campos de pesquisa, como engenharia de recursos hídricos [153], classificação de alimentos [17], reconstrução de imagens [47], reconhecimento de emoções [55], entre outros.

5.2 Metodologia

5.2.1 Banco de dados utilizados

A fim responder às perguntas estabelecidas anteriomente, um novo ciclo de experimentos foi realizado. Para essa tarefa selecionamos dezesseis bases de dados do Repositório de dados de aprendizado de máquina (UCI), descritos na Tabela 5.1.

Conjunto de dados	Nº de amostras	Nº de variáveis preditoras	Nº de classes	Tipo das variáveis
Abalone	4177	8	Numérico	Numérico, categórico
Air Quality	827	12	Numérico	Numérico
Balance Scale	625	4	3	Numérico
Biodegradation	1055	41	2	Numérico
Breast Cancer	683	9	2	Numérico
Car Evaluation	1728	6	4	Numérico
Glass Identification	214	9	6	Numérico
Haberman's Survival	306	3	2	Numérico
Heart Disease	270	13	2	Numérico
Immunotherapy	90	7	2	Numérico
Ionosphere	90	7	2	Numérico
Iris	150	4	3	Numérico
Mammographic	830	5	2	Numérico
Parkinsons	195	22	2	Numérico
WDBC	569	30	2	Numérico
Wine Quality	6497	12	2	Numérico

Tabela 5.1: Descrição das bases de dados utilizadas na segunda fase de experimentos

5.2.2 Cenários avaliados

Existem dois cenários gerais dos experimentos realizados neste capítulo: o uso de MLP e o uso de ELM para obter as matrizes de pesos para serem utilizadas como entrada para os WBFS. Nós testamos diferentes parâmetros de treinamento específicos para os modelos de cada cenário conforme a seguir:

- **Modelos baseados em MLP:** análise do número de neurônios ocultos e do valor de *weight decay*.
- **Modelos baseados em ELM:** análise do número do neurônios ocultos e da função de ativação utilizada na camada oculta e na camada de saída da RNA.

Os experimentos foram divididos em duas fases. A Fase 1 é caracterizada pelo treinamento dos modelos em diferentes cenários de parâmetros de treinamento. Os algoritmos 5.1 e 5.2 descrevem a rotina utilizada no treinamento dos dados. Duzentos modelos baseados em MLP foram treinados para cada um dos 28 pares de parâmetros $\{(hn, decay)|hn \in (5,10,15,20), decay \in (0,0.0001,0.001,0.01,0.1,0.3,0.5)\}$ e 200 modelos baseados em ELM foram treinados para cada um dos 16 pares de parâmetros $\{(hn, actFun)|hn \in (5,10,15,20), actFun \in (linear, sin, radbas, tansig)\}$, em que *sin* representa a função seno, *radbas* representa a função de ativação de base radial, e *tansig* representa a função tangente sigmoide.

Algoritmo 5.1: Treinamento dos modelos do tipo MLP

Entrada: banco de dados $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)|x \in \mathbb{R}^g\}$.

Saída: Lista contendo a performance e matriz de pesos de cada modelo

```
1 P ← {(hn, decay)|hn ∈ (5,10,15,20), decay ∈
    (0,0.0001,0.001,0.01,0.1,0.3,0.5)}.
2 for i ← 1 to 28 by 1 do
3     for j ← 1 to 200 by 1 do
4         treinar MLP(D, P_i)
5         salvar a performance
6         salvar a matriz de pesos
7     end
8 end
```

Algoritmo 5.2: Treinamento dos modelos do tipo ELM

Entrada: banco de dados $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) | x \in \mathbb{R}^g\}$.

Saída: Lista contendo a performance e matriz de pesos de cada modelo

```
1 P ← {(hn, actFun)|hn ∈ (5,10,15,20), actFun ∈
    (linear, sin, radbas, tansig)}.
2 for i ← 1 to 16 by 1 do
3     for j ← 1 to 200 by 1 do
4         treinar ELM(D, P_i)
5         salvar a performance
6         salvar a matriz de pesos
7     end
8 end
```

Cada modelo foi treinado de forma independente com diferentes inicializações dos pesos em números aleatórios e pequenos. O treinamento ocorreu por meio de validação cruzada em 10 *folds*. Ao todo foram treinados 89 600 RNAs do tipo MLP (16 bases de dados $\times$ 200 modelos $\times$ 28 pares de parâmetros de treinamento) e 51 200 RNAs do tipo ELM (16 bases de dados $\times$ 200 modelos$\times$ 16 pares de parâmetros de treinamento).

A performance dos modelos de classificação foi avaliada utilizando a acurácia e a performance dos modelos de regressão foi avaliada por meio de 1 - NRMSE, métricas descritas no capítulo anterior.

A segunda fase dos experimentos deste capítulo é caracterizada pela aplicação dos métodos WBFS em cada matriz de pesos geradas na fase 1, na aplicação da abordagem de votação, e comparação dos resultados.

Os WBFS aplicados nesta fase são os algoritmos de Garson, Olden e de Yoon, que foram os algoritmos identificados no capítulo anterior como os WBFS que obtiveram melhor desempenho. Além desses algoritmos, também aplicamos outros três novos algoritmos que são baseados no algoritmos de Garson, Olden e Yoon. A proposta destes novos algoritmos são descritos na próxima seção.

5.2.3 Proposta de melhoria dos algoritmos de Garson, Olden e Yoon

A proposta de melhoria dos algoritmos de Garson, Olden e de Yoon é baseada na combinação entre os algoritmos originais com o seletor de variáveis ReliefF. Os novos algoritmos são denominados *Feature selector based on Garson's and ReliefF algorithm (FSGR)*, *Feature Selector based on Olden's and ReliefF algorithm (FSOR)* e *Feature selector based on Yoon's and ReliefF algorithm (FSYR)*. Os algoritmos originais são

ajustados utilizando o valor de importância determinado pelo algoritmo ReliefF, conforme as equações a seguir:

$$FSGR_{ik} = \frac{\left(\sum_{j=1}^{h} \frac{|w_{ij} \cdot v_{jk}|}{\sum_{i=1}^{g} |w_{ij}|} \right) \cdot ReliefF_i}{\sum_{i=1}^{g} \left(\sum_{j=1}^{h} \frac{|w_{ij} \cdot v_{jk}|}{\sum_{i=1}^{g} |w_{ij}|} \right) \cdot ReliefF_i}, \tag{5-1}$$

$$FSOR_{ik} = \left(\sum_{j=1}^{h} w_{ij} \cdot v_{jk} \right) \cdot ReliefF_i, \tag{5-2}$$

e

$$FSYR_{ik} = \frac{\left(\sum_{j=1}^{h} w_{ij} \cdot v_{jk} \right) \cdot ReliefF_i}{\sum_{i=1}^{g} | \left(\sum_{j=1}^{h} w_{ij} \cdot v_{jk} \right) \cdot ReliefF_i|}, \tag{5-3}$$

em que $ReliefF_i$ é o valor de importância da i-ésima variável determinado pelo algoritmo ReliefF correspondente ao i-ésimo neurônio de entrada da RNA utilizada para determinar os pesos de conexão $\mathbf{w}$ e $\mathbf{v}$. O valor de $ReliefF_i$ pode variar em um espaço de -1 até +1 fornecendo um valor de importância em uma faixa de valores fixo para estimar a importância de variáveis em problemas de classificação ou de regressão [112].

Os algoritmos Relief e ReliefF

O algoritmo Relief é do tipo filtro, que estima a importância de uma variável baseado na diferença entre os valores da variável em s amostras aleatórias e seus vizinhos mais próximos das duas classes de dados analisadas [66]. Considere um conjunto de dados $\mathcal{D} = \{(\mathbf{x}_i, y_i) | i = (1,2,\cdots,n), \mathbf{x}_i \in \mathbb{R}^g\}$ com n amostras e g variáveis. O valor de importância da f-ésima variável baseada em s amostras selecionadas de maneira aleatória é dada por:

$$Relief_f^i = Relief_f^{i-1} + \frac{diff_f(x, A(x))}{s} - \frac{diff_f(x, B(x))}{s} \tag{5-4}$$

em que $i \in s$, f representa a variável a ser determinado o valor de importância, $diff_f$ representa a distância entre as amostras, $A(x)$ e $B(x)$ representam as amostras mais pró-

ximas pertencentes à mesma classe de dados e à classe de dados oposta, respectivamente. Este algoritmo é limitado a problemas de classificação binário.

Posteriormente, o algoritmo ReliefF foi proposto para lidar com problemas de classificação de múltiplas classes e com ruídos [67]. Este algoritmo realiza o cálculo da importância conforme a seguir:

$$ReliefR_f^i = ReliefF_f^{i-1} + \sum_{c \neq class(x)} \frac{\frac{p(x)}{1-(p(class(x)))} \sum_{j=1}^{k} diff_f(x, A(x))}{s \cdot l} - \sum_{j=1}^{k} \frac{diff_f(x, B(x))}{s \cdot l} \tag{5-5}$$

em que $p()$ representa a probabilidade, l é uma amostra selecionada de maneira aleatória em cada classe, e c é a classe que é diferente de $class(x)$. Posteriormente, o algorítimo ReliefF foi estendido para problemas de regressão [111].

O algoritmo ReliefF foi escolhido para melhorar os algoritmos de Garson, Olden e de Yoon devido ao seu potencial de estimar a importância das variáveis em problemas de classificação e de regressão com dependências fortes entre as variáveis [112]. Ademais, nós queríamos adicionar uma informação diretamente relacionada ao conjunto de treinamento nos métodos WBFS considerando que estes métodos utilizam apenas os pesos da RNA para calcular a importância das variáveis.

Detalhes de implementação e uso

Os modelos de RNA codificam variáveis categóricas em novas variáveis binárias para treinar dados qualitativos. No entando, esse processo não ocorre com o algoritmo ReliefF. Dessa forma, as variáveis de entrada de uma RNA se diferenciam das variáveis de entrada para o ReliefF.

Nesse sentido, é necessário realizar a codificação das variáveis categóricas em novas variáveis binárias, e utilizar este conjunto de dados como entrada de dados para a RNA e para o algoritmo ReliefF para garantir que o peso de conexão do i-ésimo neurônio de entrada corresponda com o valor de importância da i-ésima variável segundo o ReliefF.

5.2.4 Hipóteses e testes

Cinco hipóteses foram estabelecidas com o objetivo de investigar o comportamento dos WBFS em face a diferentes modelos de RNA baseadas em diferentes parâmetros de treinamento. Ademais, a estabilidade dos algoritmos de Garson, Olden, Yoon, e suas melhorias FSGR, FSOR e FSYR também foram avaliadas.

As hipóteses deste capítulo são:

Hipótese 1. Existe uma diferença significativa entre as estabilidades dos WBFS obtidas por diferentes parâmetros de treinamento da RNA.
Hipótese 2. Existe uma diferença significativa entre as estabilidades dos WBFS obtidas por RNAs baseadas em MLP e RNAs baseadas em ELM.
Hipótese 3. Existe uma diferença significativa entre as medidas de estabilidade a depender do WBFS utilizado.
Hipótese 4. Existe uma diferença significativa no ranking final de importância a depender do WBFS utilizado.
Hipótese 5. A ponderação dos WBFS por alguma métrica de importância melhora a estabilidade do WBFS.

A verificação das Hipóteses 1, 3 e 5 foram realizadas por meio dos testes estatísticos de Friedman e de Nemenyi, visto que a comparação realizada nessas hipóteses envolve mais de dois grupos em amostras pareadas. Para a Hipótese 2 nós utilizamos o teste de classificação sinalizada de Wilcoxon (*Wilcoxon signed-rank test*), uma vez que a comparação realizada para avaliar a hipótese é a comparação de apenas dois algoritmos em amostras pareadas. Por fim, a Hipótese 4 foi analisada por meio do coeficiente de correlação de Spearman .

O coeficiente de correlação de Spearman (S_r) é a métrica mais utilizada para medir a estabilidade entre dois rankings de importância [116]. Considere um conjunto de diferentes rankings de importância $\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_1, \cdots, \mathbf{r}_K\}$. O resultado médio das similaridades pareadas é dada por:

$$\Phi(\mathcal{R}) = \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} S_r(r_i, r_j) \tag{5-6}$$

em que K é o número de diferentes rankings contidos em $\mathcal{R}$. O coeficiente de correlação S_r entre dois rankings compostos de g variáveis é definido por:

$$S_r(p, p') = 1 - 6 \sum_{i=1}^{g} \frac{(p_i - p'_j)^2}{g(g^2-1)}. \tag{5-7}$$

5.3 Resultados

Nesta seção, apresentamos os resultados de diferentes experimentos para avaliar a estabilidade dos WBFS em diferentes cenários de parâmetros de treinamento e dois modelos diferentes de RNA. Primeiro, são apresentados os resultados do desempenho da classificação entre os modelos de RNA nos diferentes cenários de parâmetros de treinamento. Em seguida, as medidas de estabilidade são comparadas e o melhor cenário é selecionado para comparar a estabilidade dos modelos do tipo MLP e ELM. Em seguida,

os algoritmos WBFS foram comparados com base no melhor cenário em termos de medidas de estabilidade da abordagem de votação. Por fim, a quantidade de modelos necessários para atingir a estabilidade é avaliada, assim como a robustez dos rankings obtidos.

5.3.1 Desempenho dos modelos MLP e ELM nos diferentes cenários

Os diferentes cenários de parâmetros de treinamento resultaram em diferentes capacidades de predição (Figuras 5.1 e 5.2). Nos mapas de calor, cada coluna representa o número de neurônios ocultos e cada linha representa os valores de *weight decay* para os modelos MLP (Figura 5.1), e cada linha representa as funções de ativação para os modelos ELM (Figuras 5.2). A intensidade das caixas coloridas representam a média de desempenho dos 200 modelos de RNA em cada par de parâmetros de treinamento. Os resultados do teste de Friedman indicaram que há uma diferença estatisticamente significativa entre os desempenhos obtidos de diferentes parâmetros (χ^2 = 139,37, valor p <2,2e-16) para os desempenhos MLP, e também para as performances dos modelos baseados em ELM (χ^2 = 128,14, valor p <2,2 e-16).

As Figuras 5.3 e 5.4 apresentam os resultados *post-hoc* de Nemenyi na forma do diagrama de diferença crítica (CD, *critical difference*). As figuras apresentam o intervalo CD na linha horizontal superior esquerda e o ranking médio no eixo x. Nessa figura, os cenários de parâmetros que obtiveram os melhores desempenhos são colocados no lado esquerdo (rankings médios menores). Os cenários de parâmetros estatisticamente equivalentes são conectados por uma barra. A partir desses diagramas, é possível observar que os melhores desempenhos de MLP foram obtidos pelos maiores valores de *weight decay*, sendo estes estaticamente equivalentes aos menores valores de *weight decay*, porém, com um ranking médio maior. Ademais, os melhores desempenhos dos modelos baseados em ELM foram obtidos pela função de ativação *linear*, possuindo os menores rankings médios e sendo estaticamente equivalentes a função de ativação *tansig* na maioria dos casos. Além disso, há uma diferença estatística significativa entre os desempenhos do algoritmo MLP e ELM, em que o MLP forneceu as maiores precisões (valor p < 2,2e-16).

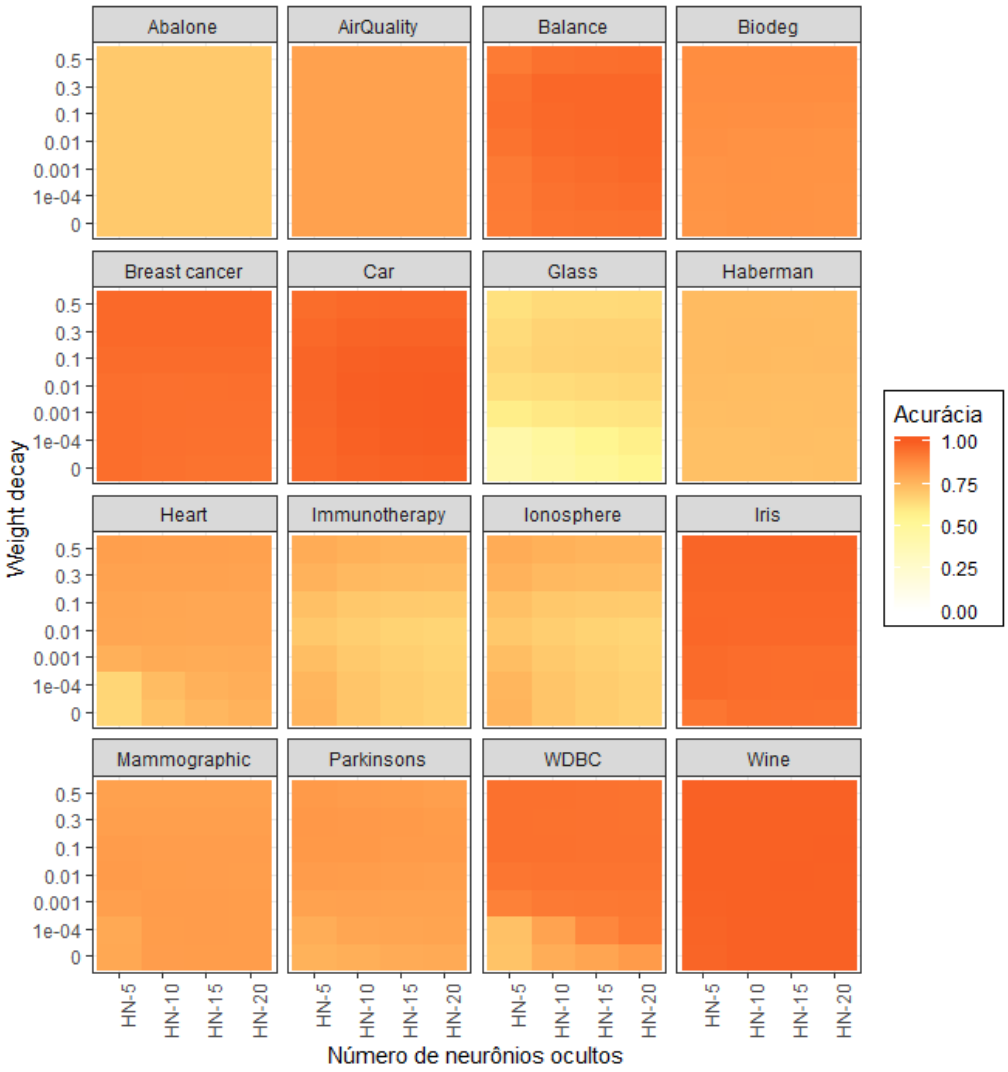


Figura 5.1: Capacidade de predição para cada conjunto de dados em cada cenário de treinamento dos modelos baseados em MLP.

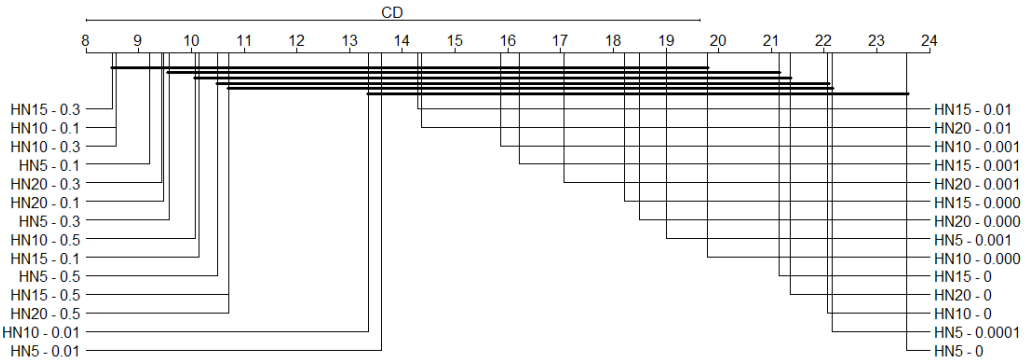


Figura 5.3: Diagrama de diferença crítica para o desempenho dos modelos do tipo MLP para cada cenário de parâmetros de treinamento.

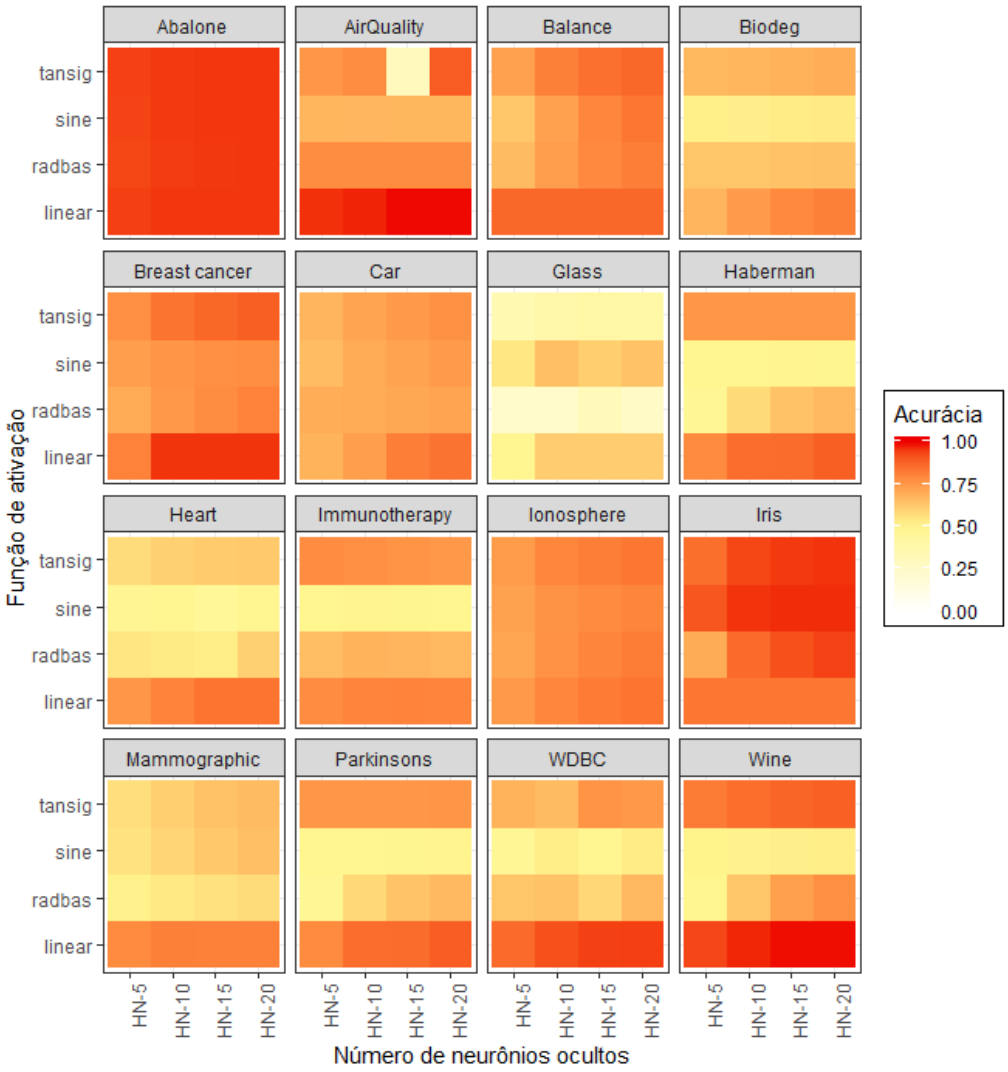


Figura 5.2: Capacidade de predição para cada conjunto de dados em cada cenário de treinamento dos modelos baseados em ELM.

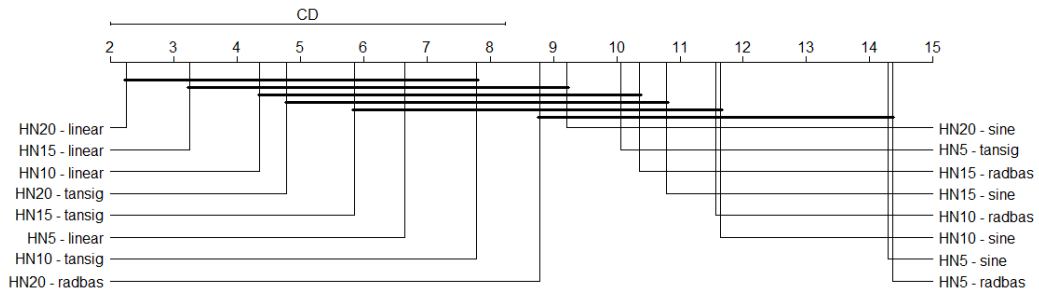
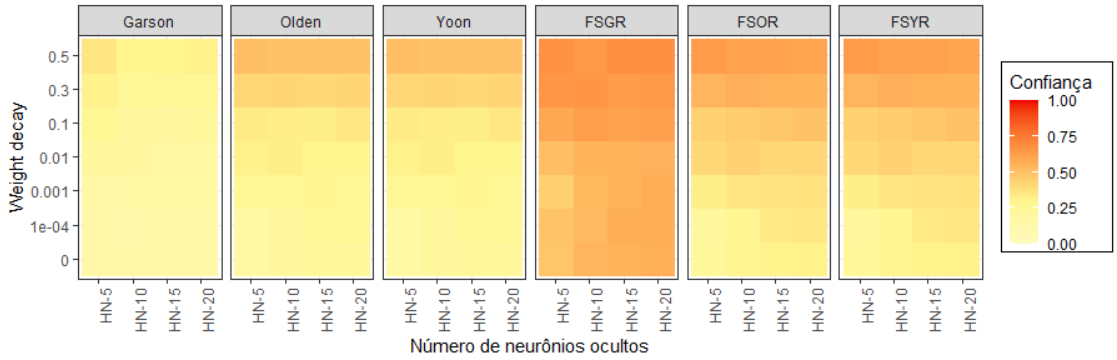


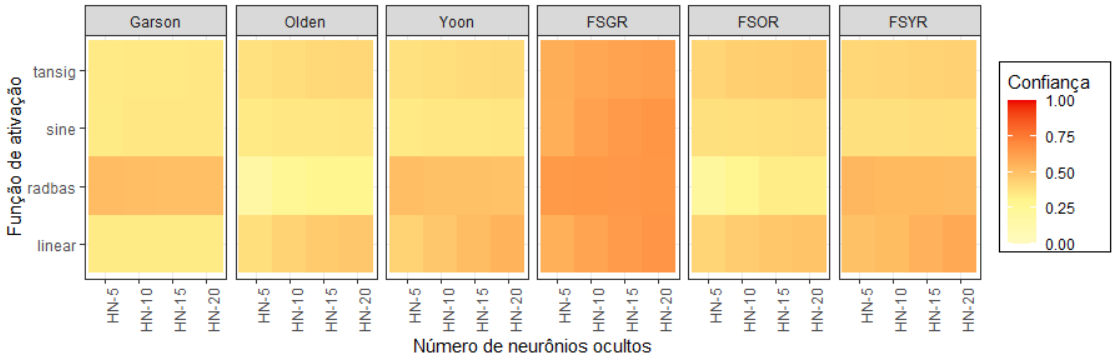
Figura 5.4: Diagrama de diferença crítica para o desempenho dos modelos do tipo ELM para cada cenário de parâmetros de treinamento.

Nos resultados do próximo experimento, apresentaremos e discutiremos a estabilidade dos WBFS obtidos a partir dos diferentes cenários de parâmetros. Até onde sabemos, não há evidências de que os modelos de RNA com os maiores desempenhos resultem nas melhores medidas de estabilidade. Posteriormente, confrontamos os resultados citados anteriormente com os resultados dos quais o cenário de parâmetros resulta nas melhores medidas de estabilidade.

As Figuras 5.5 e 5.6 apresentam os valores médios de confiança e da quantidade de posições votadas, respectivamente, para cada WBFS nos 16 pares de parâmetros de treinamento ELM e os 28 pares de parâmetros de treinamento MLP, considerando todos os conjuntos de dados. É possível observar uma variação nos valores de confiança, em que o algoritmo FSGR parece resultar nos maiores valores de confiança (tons mais escuros na Figura 5.5) e na menor quantidade de posições votadas (tons mais claros na Figura 5.6).

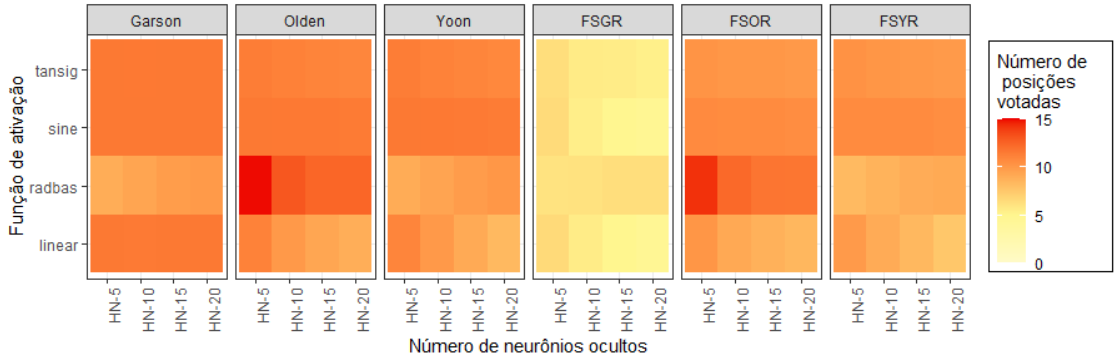


(a) Modelos baseados em MLP

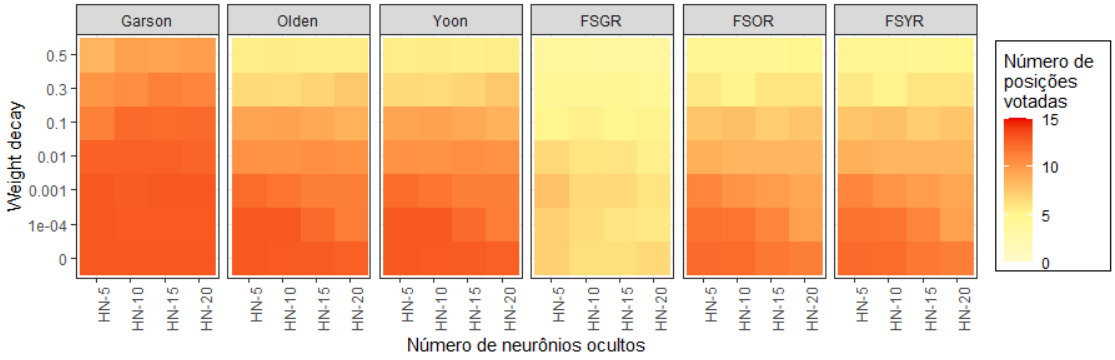


(b) Modelos baseados em ELM

Figura 5.5: Mapa de calor para a média dos valores de confiança obtidos nos experimentos considerando todos os conjuntos de dados.



(a) Modelos baseados em MLP



(b) Modelos baseados em ELM

Figura 5.6: Mapa de calor para média da quantidade de posições votadas obtidas nos experimentos considerando todos os conjuntos de dados.

Para atestar esses resultados, nas próximas subseções analisaremos a diferença estatística entre os cenários de parâmetros e determinaremos o melhor cenário para cada algoritmo para ambas as abordagens MLP e ELM. Em seguida, compararemos a estabilidade do WBFS considerando o melhor cenário para cada um.

5.3.2 Determinado o melhor cenário de parâmetro de treinamento

Aplicamos o teste de Friedman seguido do teste *post-hoc* de Nemenyi para cada WBFS para avaliar as diferenças entre os cenários de parâmetros. Primeiro aplicamos o teste estatístico para investigar as diferenças entre o número de neurônios ocultos para cada valor de *weight decay* nos modelos MLP e para cada função de ativação nos modelos ELM. Em seguida, aplicamos o teste estatístico para investigar as diferenças entre os valores de *weight decay* e as diferentes funções de ativação para cada número de neurônios ocultos. Essa separação foi projetada para evitar resultados enviesados quanto à possibilidade de diferenças no número de neurônios ocultos interferirem na análise do

valor de *weight decay* e das funções de ativação, situação válida também para o caso contrário.

A Tabela 5.2 apresenta os resultados do teste de Friedman para analisar as diferenças nos valores de confiança considerando os valores de *weight decay* para cada número de neurônios ocultos em cada WBFS. Os resultados da análise das diferenças no número de neurônios ocultos são identificados pelas notas de rodapé com nível de significância de 0,05. A hipótese nula (H_0) nesse estágio do estudo foi: "Não há diferenças significativas entre os resultados obtidos de diferentes parâmetros de treinamento"; e a hipótese alternativa (H_1) é o oposto. Os valores de p menores que 0,05 rejeitam a hipótese nula (H_0) e mostram que há uma diferença significativa dependendo do valor de *weight decay* usado para treinar o modelo MLP. Essa diferença nos valores de *weight decay* foi encontrada para todos os número de neurônios ocultos, em todos os algoritmos WBFS estudados. Em seguida nós aplicamos o teste *post-hoc* de Nemenyi para identificar os valores de *weight decay* que resultaram nas maiores estabilidades.

Tabela 5.2: Resultado do teste de Friedman para comparar as diferenças entre os valores de *weight decay* nos modelos baseados em MLP.

WBFS	Nº de neurônios ocultos	χ^2	valor p	Rejeitar H_0 ?
Garson	HN-5	73,98	<0,001	Sim
	HN-10	69,98	<0,001	Sim
	HN-15	72,16	<0,001	Sim
	HN-20	81,63	<0,001	Sim
FSGR	HN-5	83,46	<0,001	Sim
	HN-10	71,21	<0,001	Sim
	HN-15[g,h,i]	77,83	<0,001	Sim
	HN-20[a,b,c,d,e,f]	77,47	<0,001	Sim
Olden	HN-5	60,87	<0,001	Sim
	HN-10	53,22	<0,001	Sim
	HN-15[h]	59,94	<0,001	Sim
	HN-20[b]	62,43	<0,001	Sim
FSOR	HN-5	55,30	<0,001	Sim
	HN-10	44,09	<0,001	Sim
	HN-15	50,30	<0,001	Sim
	HN-20[b]	52,44	<0,001	Sim
Yoon	HN-5	69,60	<0,001	Sim
	HN-10	62,62	<0,001	Sim
	HN-15[g,h,i]	68,82	<0,001	Sim
	HN-20[a,b,c,e]	72,28	<0,001	Sim
FSYR	HN-5	79,51	<0,001	Sim
	HN-10	65,91	<0,001	Sim
	HN-15[h]	75,49	<0,001	Sim
	HN-20[a,b,c,e]	75,36	<0,001	Sim

[a] Não há diferença significativa entre HN-20 e HN-5 com *weight decay* = 0.
[b] Não há diferença significativa entre HN-20 e HN-5 com *weight decay* = 0,0001.
[c] Não há diferença significativa entre HN-20 e HN-5 com *weight decay* = 0,001.
[d] Não há diferença significativa entre HN-20 e HN-10 com *weight decay* = 0.
[e] Não há diferença significativa entre HN-20 e HN-10 com *weight decay* = 0,0001.
[f] Não há diferença significativa entre HN-20 e HN-10 com *weight decay* = 0,001.
[g] Não há diferença significativa entre HN-15 e HN-5 com *weight decay* = 0.
[h] Não há diferença significativa entre HN-15 e HN-5 com *weight decay* = 0,0001.
[i] Não há diferença significativa entre HN-15 e HN-5 com *weight decay* = 0,001.

A Figura 5.7 apresenta o diagrama de diferença crítica para cada WBFS em cada número de neurônios ocultos. Nesta figura, os melhores valores de confiança são

colocados no lado esquerdo (rankings médios menores). Os valores de *weight decay* que são estatisticamente equivalentes são conectados por uma barra preta.

Os maiores valores de *weight decay* (0,5, 0,3, 0,1) são estatisticamente equivalentes em 100% dos casos, sendo que o valor de 0,5 é o que sempre resultou no menor ranking médio. Em todos os casos, este valor é significativamente diferente dos valores de *weight decay* (0, 0,0001, 0,001). Esses valores resultaram nos rankings médios mais altos. Em 37,5% dos casos, o valor de *weight decay* 0,01 também é equivalente a (0,5, 0,3, 0,1). Os menores valores de *weight decay* (0, 0,0001, 0,001) são estatisticamente equivalentes em 100% dos casos, e em 75% dos casos esses valores de *weight decay* também são estatisticamente equivalentes a (0,01). Apenas para valores de *weight decay* mais baixos foram encontradas diferenças significativas entre o número de neurônios ocultos 20 e 5, 20 e 10 e 15 e 20.

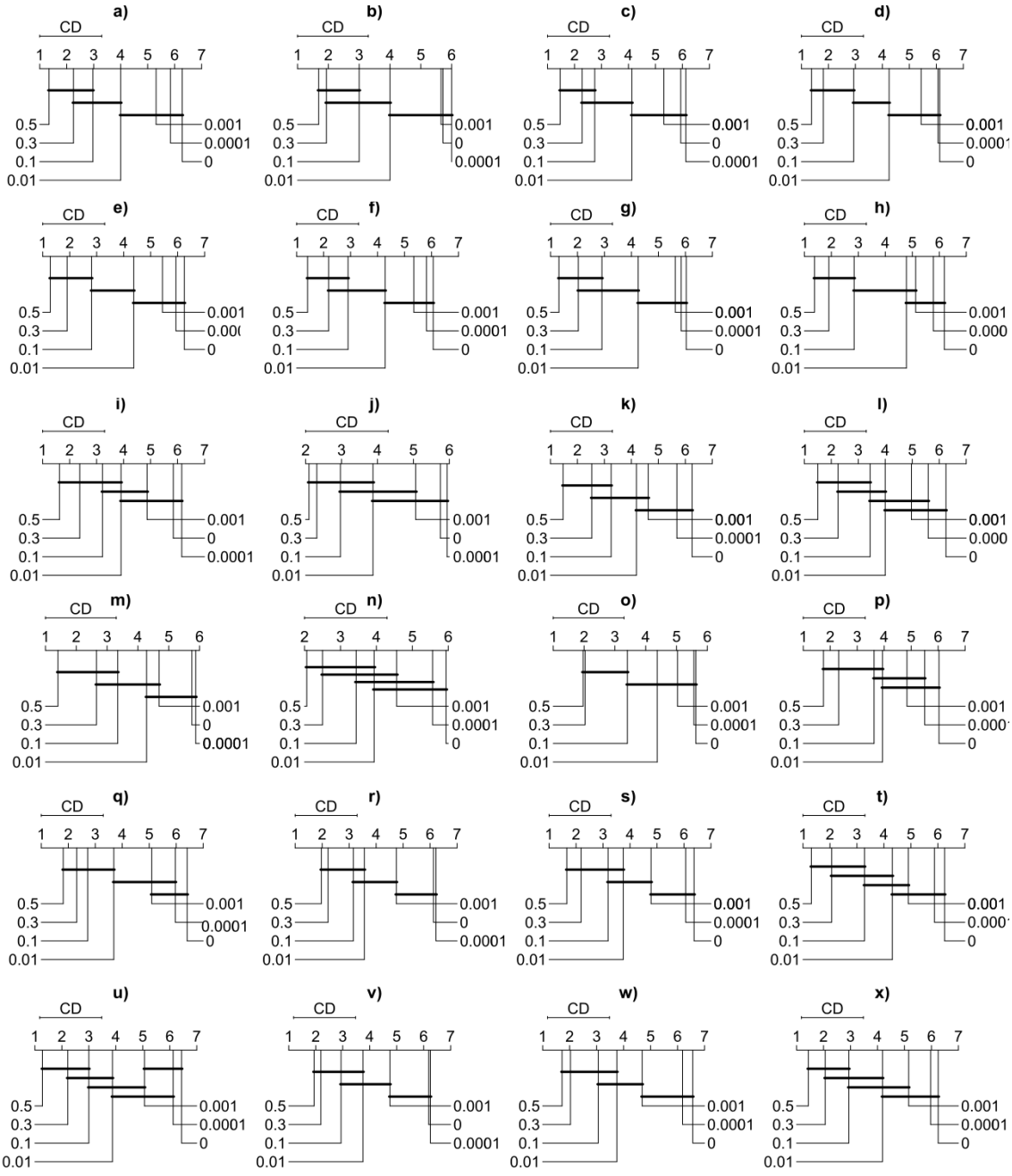


Figura 5.7: Teste de Nemenyi para os modelos MLP com 5, 10, 15 e 20 neurônios ocultos para o algoritmo de Garson (a-d), FSGR (e-h), Olden (i-l), FSOR (m-p), Yoon (q-t) e FSYR (u-x).

Esses resultados indicam que, entre os parâmetros testados, valores de *weight decay* mais altos fornecem valores de confiança mais altos. Valores maiores de *weight decay* tendem a reduzir os pesos para zero [38]. Nenhuma diferença significativa entre o número de neurônios ocultos foi encontrada em valores de *weight decay* mais elevados (Tabela 5.2, notas de rodapé).

No início da seção de Resultados descrevemos os resultados da capacidade de predição dos modelos. Foi identificado que os maiores valores de *weight decay* resulta-

ram nos menores rankings médios no teste de Friedman. No entanto, a performance obtida pelos maiores valores de *weight decay* foi estatisticamente equivalente à performance obtida por modelos treinados com os menores valores de *weight decay*. Dessa forma, observamos que os maiores valores de *weight decay* resultam em confianças estatisticamente maiores do que as obtidas por velores menores de *weight decay*, enquanto o mesmo não é válido para a performance dos modelos estudados neste capítulo.

A Tabela 5.3 apresenta os resultados do teste de Friedman para analisar as diferenças nos valores de confiança considerando as diferentes funções de ativação no modelo ELM para cada número de neurônios ocultos para cada WBFS. Os resultados da análise das diferenças entre o número de neurônios ocultos foram identificados nas notas de rodapé. Os valores de p menores que 0,05 rejeitam a hipótese nula H_0 e mostram que há uma diferença significativa dependendo da função de ativação usada para treinar o modelo ELM. Essa diferença foi encontrada para todos os números de neurônios ocultos em todos os algoritmos, exceto para os modelos com 10, 15 e 20 neurônios ocultos no algoritmo de Garson e os modelos com 5 neurônios ocultos no algoritmo FSGR. Para identificar em quais funções de ativação há uma diferença significativa, aplicamos o teste *post-hoc* de Nemenyi.

Tabela 5.3: Resultado do teste de Friedman para comparar as diferenças entre as funções de ativação nos modelos baseados em ELM.

WBFS	Nº de neurônios ocultos	χ^2	valor p	Rejeitar H_0 ?
Garson	HN-5	9,35	0,0249	Sim
	HN-10	1,05	0,7880	Não
	HN-15	1,69	0,6375	Não
	HN-20[a]	5,88	0,1175	Não
FSGR	HN-5	3,79	0,285	Não
	HN-10	24,23	<0,001	Sim
	HN-15[g,h]	16,29	<0,001	Sim
	HN-20[a,b,c,d,e,f]	22,38	<0,001	Sim
Olden	HN-5	23,45	<0,001	Sim
	HN-10	14,42	0,0023	Sim
	HN-15[i]	14,53	0,0022	Sim
	HN-20[d]	16,34	<0,001	Sim
FSOR	HN-5	22,93	<0,001	Sim
	HN-10	23,67	<0,001	Sim
	HN-15[i]	19,27	<0,001	Sim
	HN-20	19,06	<0,001	Sim
Yoon	HN-5	18,23	<0,001	Sim
	HN-10	24,39	<0,001	Sim
	HN-15[i]	26,81	<0,001	Sim
	HN-20[d]	21,27	<0,001	Sim
FSYR	HN-5	21,92	<0,001	Sim
	HN-10	28,29	<0,001	Sim
	HN-15[i]	18,12	<0,001	Sim
	HN-20	27	<0,001	Sim

[a] Não há diferença significativa entre HN-20 e HN-5 com a função de ativação seno.

[b] Não há diferença significativa entre HN-20 e HN-5 com a função de ativação linear.

[c] Não há diferença significativa entre HN-20 e HN-5 com a função de ativação de base radial.

[d] Não há diferença significativa entre HN-20 e HN-5 com a função de ativação tangente sigmoide.

[e] Não há diferença significativa entre HN-20 e HN-10 com a função de ativação seno.

[f] Não há diferença significativa entre HN-20 e HN-10 com a função de ativação linear.

[g] Não há diferença significativa entre HN-15 e HN-5 com a função de ativação seno.

[h] Não há diferença significativa entre HN-15 e HN-5 com a função de ativação linear.

[i] Não há diferença significativa entre HN-15 e HN-5 com a função de ativação tangente sigmoide.

A Figura 5.8 apresenta o diagrama de diferenças críticas para cada WBFS em cada número diferente de neurônios ocultos considerando os modelos ELM. As funções de ativação estatisticamente equivalentes foram conectadas por uma barra preta. O teste de Friedman resultou em um valor de p de 0,0249 para os modelos com 5 neurônios ocultos com o algoritmo de Garson (Tabela 5.3), indicando que existem ao menos dois algoritmos com valores estatisticamente diferentes. No entanto, o teste *post-hoc* de Nemenyi não identificou diferenças significativas entre as funções de ativação para os modelos com 5, 10, 15 ou 20 para o algoritmo de Garson (Figura 5.8, subfiguras “a)”,“b)”,“c)” e“d)”).

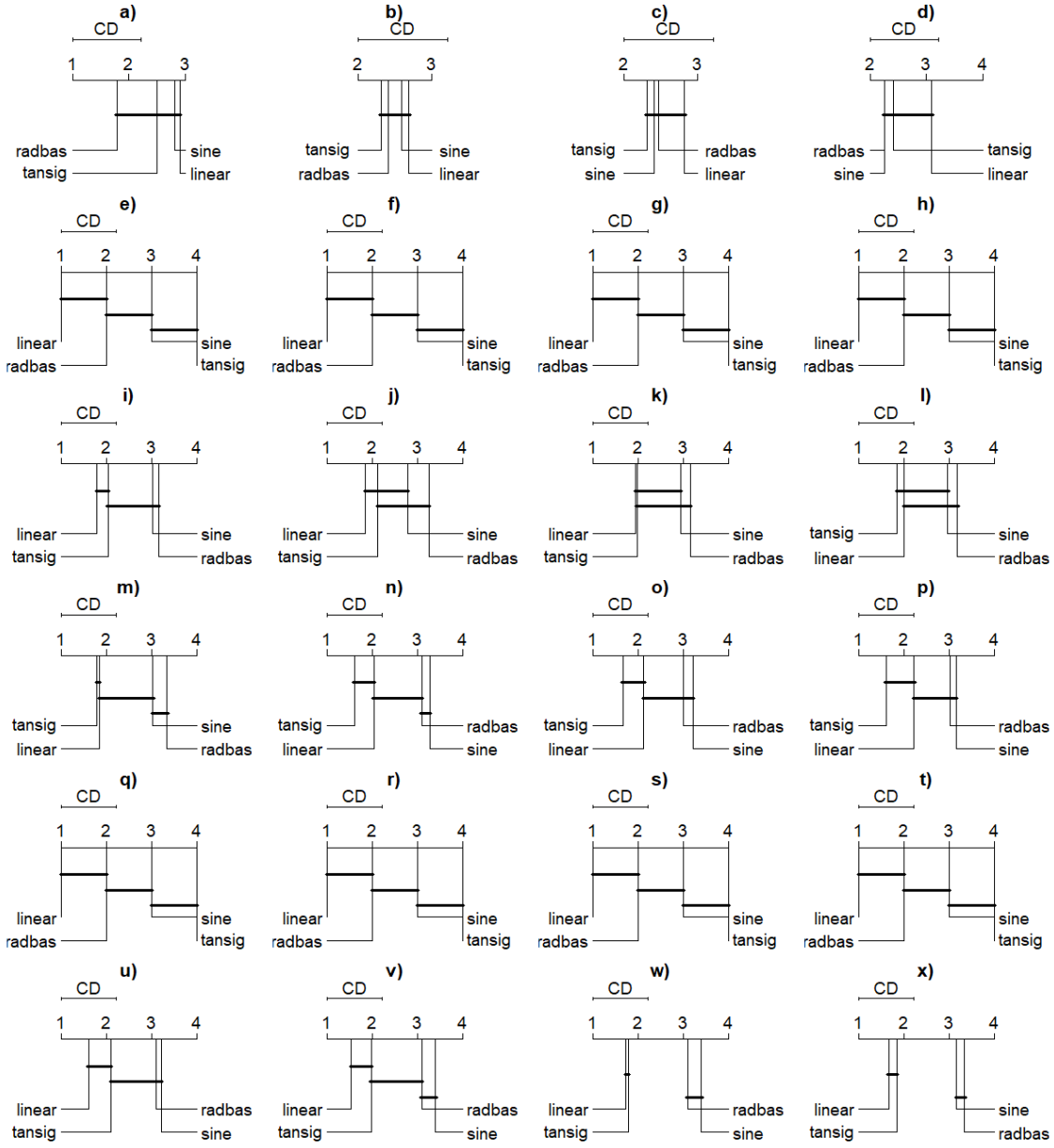


Figura 5.8: Teste de Nemenyi para os modelos ELM com 5, 10, 15 e 20 neurônios ocultos para o algoritmo de Garson (a-d), FSGR (e-h), Olden (i-l), FSOR (m-p), Yoon (q-t) e FSYR (u-x).

As funções de ativação linear e tan-sigmoide são estatisticamente equivalentes em 100% dos casos em que o teste de Nemenyi identificou diferenças significativas. Em 73,68% dos casos, a função de ativação linear foi a que obteve o ranking médio mais baixo e teve diferenças significativas em relação à função de base radial. A função seno foi significativamente diferente da melhor função de ativação (linear ou tan-sigmoide) em 63,15% dos casos. As quatro funções de ativação tiveram diferenças significativas em pelo menos um dos seguintes pares de neurônios ocultos: 20 e 5, 20 e 10, e 15 e 20. Os neurônios ocultos 15 e 20 foram estatisticamente equivalentes em todos os casos (Tabela 5.3, notas de rodapé). Os resultados da análise da quantidade de posições votadas seguem o mesmo padrão que o resultado da análise dos valores de confiança (dados não mostrados).

Em suma, com base nos resultados obtidos nesta subseção, podemos afirmar que a estabilidade de um WBFS varia de acordo com os parâmetros usados para treinar a RNA. O número de neurônios ocultos afeta as medidas de estabilidade apenas algumas vezes, enquanto o valor de *weight decay* e as funções de ativação afetam as medidas de estabilidade na maioria das vezes. Considerando esses resultados, os cenários de parâmetros que alcançaram os maiores valores de confiança são apresentados na Tabela 5.4.

Tabela 5.4: Melhores cenários de parâmetros de treinamento das redes MLP e ELM

Tipo	Parâmetro	Melhor cenário
MLP	Número de neurônios ocultos	5,10,15 e 20
	Valor de *weight decay*	0,5
ELM	Número de neurônios ocultos	15 e 20
	Função de ativação	Linear e tangente sigmoide

5.3.3 Comparação dos WBFS nos melhores cenários

Nesta seção, comparamos a estabilidade dos WBFS nos modelos MLP e ELM nos melhores cenários de parâmetros de treinamento estabelecidos na seção anterior. O teste de Friedman resultou em uma diferença significativa entre os valores de confiança dos WBFS com base nos modelos MLP (χ^2 = 18,02, valor p = 0,002703), e também para a quantidade de indicações (χ^2 = 34,68, valor p = 1,7e-06). A Figura 5.9 apresenta o *boxplot* dos valores de confiança da quantidade de posições votadas para cada WBFS obtido a partir de modelos de MLP e os resultados do teste *post hoc* de Nemenyi. Embora nenhuma diferença entre o número de neurônios ocultos tenha sido encontrada no valor de *weight decay* de 0,5, usamos apenas os dados gerados com 15 e 20 neurônios ocultos para

emparelhar as amostras MLP com as amostras ELM para os experimentos das próximas seções.

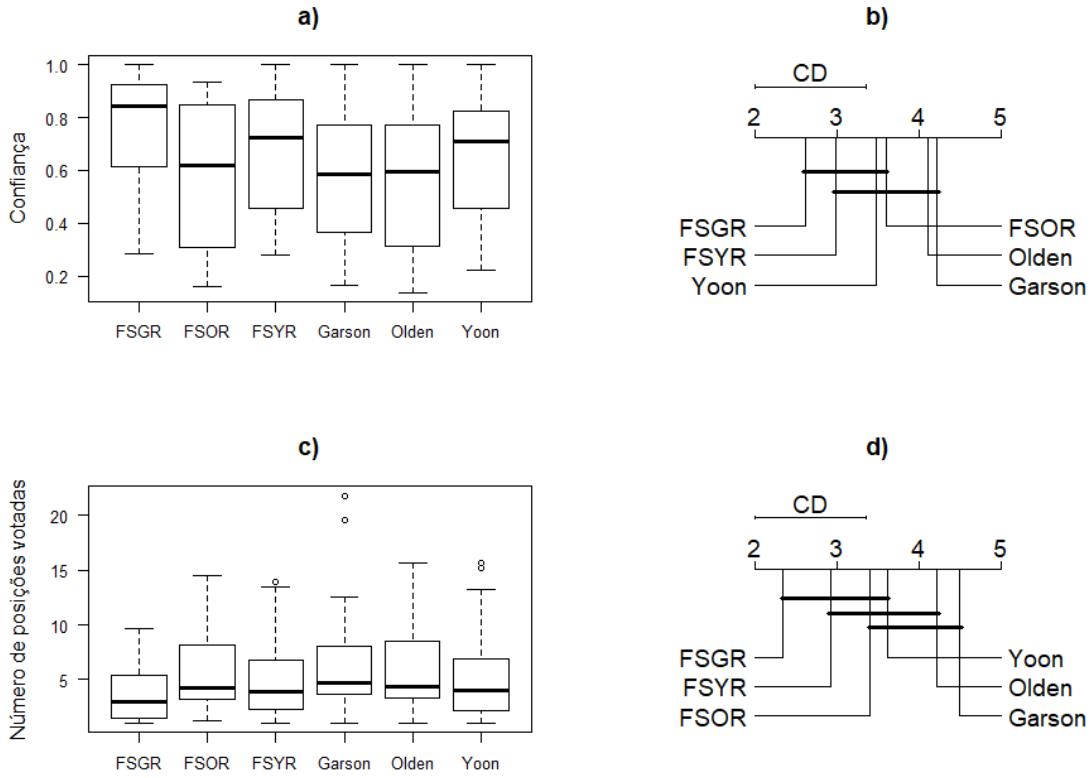


Figura 5.9: Resultado do teste de Nemenyi para comparar os WBFS no melhor cenário de parâmetros de treinamento dos modelos do tipo MLP.

De acordo com o teste *post hoc* de Nemenyi, os algoritmos propostos FSGR, FSYR, FSOR e o algoritmo de Yoon são estatisticamente equivalentes, em que o FSGR resultou no ranking médio mais baixo (Figura 5.9 (b)). O algoritmo de Garson resultou no ranking médio mais alto, indicando que este algoritmo forneceu os valores de confiança mais baixos. Os resultados apresentados na Figura 5.9 (d) para a análise da quantidade de posições votadas são semelhantes aos obtidos a partir da análise dos valores de confiança.

O teste de Friedman para a comparação dos WBFS baseados em modelos do tipo ELM também resultou em uma diferença significativa para os valores de confiança (χ^2 = 66,24, valor p = 6,1e-13) e também para a quantidade de posições votadas (χ^2 = 51,28, valor p = 7,5e-10). Os resultados do teste *post-hoc* de Nemenyi foram semelhantes aos obtidos na análise de MLP (Figura 5.10). O algoritmo proposto FSGR atingiu o ranking médio mais baixo e é estatisticamente equivalente ao FSYR, FSOR e ao algoritmo de Yoon para os valores de confiança (Figura 5.10 (b)). Considerando a quantidade de posições votadas, os melhores resultados foram obtidos a partir dos algoritmos propostos FSGR, FSOR, FSYR, e do algoritmo de Yoon e de Olden (Figura 5.10 (d)). O algoritmo

de Garson obteve o ranking médio mais alto, indicando que esse algoritmo resultou nos valores de confiança mais baixos e na maior quantidade de posições votadas.

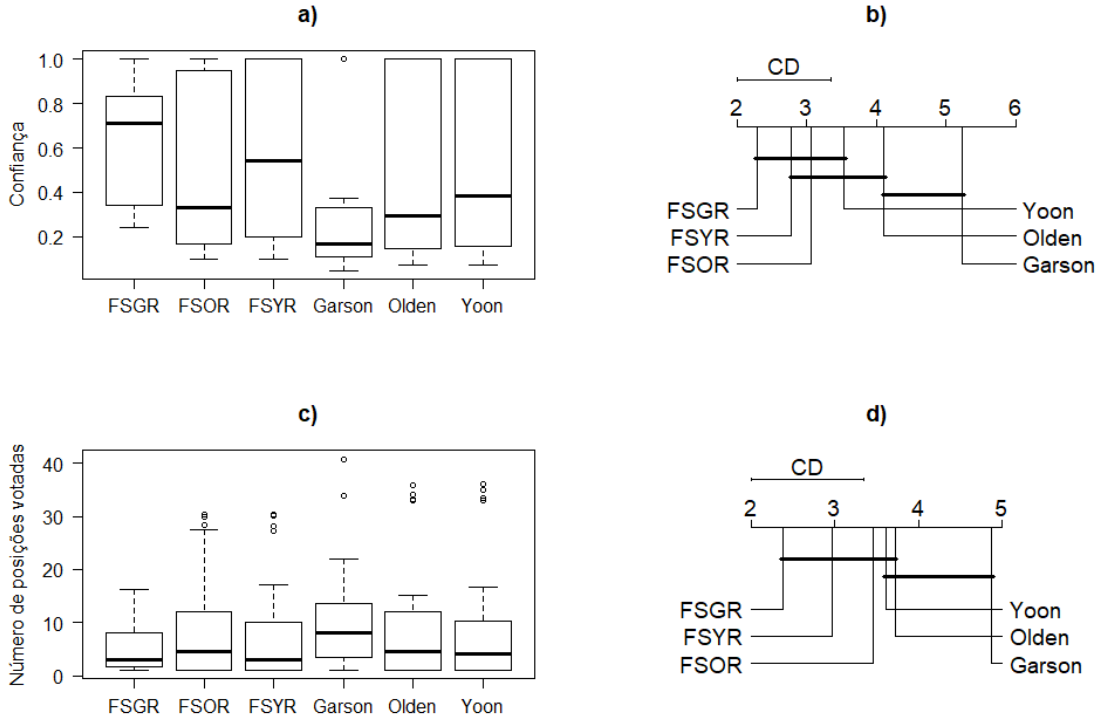


Figura 5.10: Resultado do teste de Nemenyi para comparar os WBFS no melhor cenário de parâmetros de treinamento dos modelos do tipo ELM

5.3.4 Comparação entre os WBFS baseados em MLP e ELM

Os resultados obtidos a partir dos modelos MLP e ELM em cada WBFS são considerados nesta seção. O objetivo desta análise é investigar se o custo computacional adicional do treinamento de *backpropagation* do MLP melhora significativamente a estabilidade do WBFS, assim como identificar quais técnicas de WBFS foram mais beneficiadas com um modelo de RNA diferente. Por uma questão de simplicidade, foram considerados apenas os resultados obtidos no melhor cenário para cada técnica (ou seja, os resultados apresentados nas Figuras 5.9 e 5.10).

Usamos o teste sinalizado de Wilcoxon. A hipótese nula H_0 é: "Não há diferenças significativas entre os resultados dos dois modelos de RNA" e a hipótese alternativa H_1 é o oposto. Os resultados numéricos são apresentados na Tabelas 5.5 e 5.6, em que R^+ é a soma dos rankings em que o MLP superou os resultados do ELM, R^- é a soma dos rankings no caso oposto, N^+ é a soma das vezes que o MLP teve melhor desempenho do que ELM, e N^- é a soma das vezes para o caso oposto. A soma de N^+ e N^- é 32, caracterizando os 16 bases de dados com 15 e 20 neurônios ocultos no melhor valor de

weight decay e melhor função de ativação. Os casos em que a soma de N^+ e N^+- é menor 32 indica que houve empates na comparação. Os valores de p menores que 0,05 rejeitam a hipótese nula H_0 e mostram que os resultados obtidos pelo modelo MLP foram melhores do que os obtidos pelo modelo ELM (Tabelas 5.5 e 5.6).

Tabela 5.5: Resultados do teste de Wilcoxon para os valores de confiança.

MLP vs. ELM	R^+	R^-	N^+	N^-	p-value	Rejeitar H_0?
Garson	413	115	24	8	0,0026	Sim
FSGR	408	120	26	6	0,0035	Sim
Olden	339	185	20	9	0,0749	Não
FSOR	338	190	20	12	0,0818	Não
Yoon	331	193	17	12	0,0984	Não
FSYR	308.75	217,75	18	12	0,1974	Não

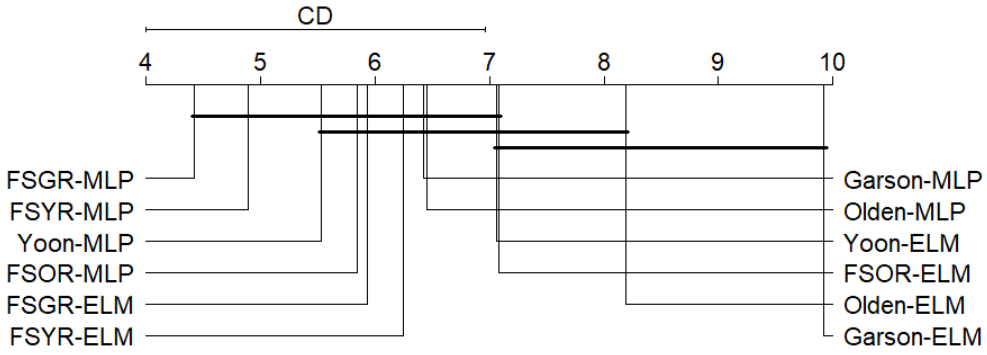
Tabela 5.6: Resultados do teste de Wilcoxon para o número de posições votadas.

MLP vs. ELM	R^+	R^-	N^+	N^-	p-value	Rejeitar H_0?
Garson	417,75	108,75	24	6	0,0019	Sim
FSGR	388,75	137,25	22	8	0,0097	Sim
Olden	328,5	187,5	18	9	0,0937	Não
FSOR	336,5	179,5	18	9	0,0701	Não
Yoon	328,5	187,5	17	10	0,0937	Não
FSYR	329	195	17	12	0,1051	Não

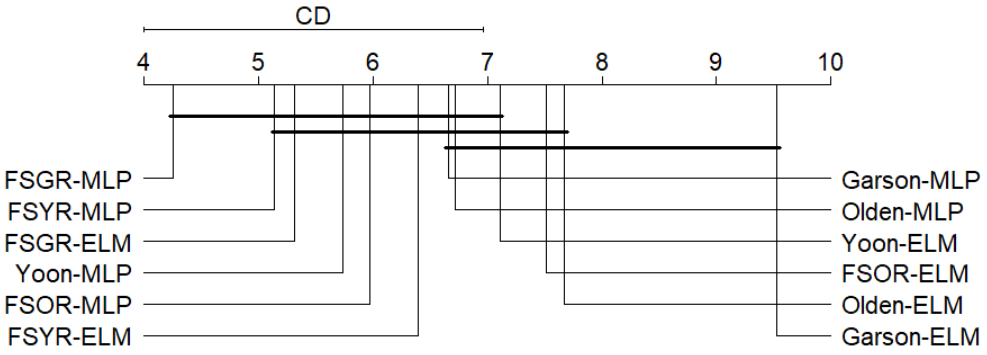
Os modelos baseados em MLP forneceram uma estabilidade significativamente maior do que os modelos baseados em ELM apenas para os algoritmos de Garson e no FSGR (valor $p < 0{,}05$). Um fato interessante é que, apesar de haver uma diferença significativa entre os desempenhos dos algoritmos MLP e ELM nos algoritmos de Garson e no FSGR, não foram encontradas diferenças para os algoritmos de Olden, de Yoon, e para os algoritmos propostos FSOR e FSYR, indicando que utilizar MLP ou ELM como base para esses algoritmos não implica em perda de desempenho.

Além do teste anterior, comparamos os resultados obtidos de todos os WBFS baseados nos modelos MLP e ELM ao mesmo tempo. O teste de Friedman resultou em uma diferença significativa entre os algoritmos para os valores de confiança (χ^2 = 61,86, valor p = 34,1e-09) e também para a quantidade de posições votadas (χ^2 = 58,02, valor p = 2,1 e-08). Embora apenas o algoritmo FSGR dentre os algoritmos propostos tenha resultado em melhores medidas de estabilidade a partir de modelos baseados em MLP do que modelos baseados em ELM, ao considerar os demais WBFS baseados em MLP e

ELM na mesma análise, todos os algoritmos propostos baseados em MLP e ELM foram estatisticamente equivalentes (Figura 5.11, algoritmos FSGR, FSOR e FSYR baseados em MLP e em ELM são conectados por uma barra).



(a) Resultados para a análise dos valores de confiança.



(b) Resultados para a análise do número de posições votadas.

Figura 5.11: Resultado do teste de Nemenyi considerando os WBFS para os dois tipos de RNA estudadas.

Este resultado indica a possibilidade de aplicar um dos algoritmos FSGR, FSYR, FSOR, e o algoritmo de Yoon a dados de alta dimensão usando ELM em vez de MLP com *backpropagation*. O uso de WBFS baseado em modelos de RNA treinados com *backpropagation* em dados de alta dimensão requer uma demanda computacional maior em comparação com o algoritmo ELM. Isso ocorre devido à necessidade de treinar um grande conjunto de modelos de RNA para calcular a estabilidade de WBFS.

Como resultado desses experimentos, FSGR foi o algoritmo WBFS que forneceu a maior estabilidade, seguido por FSOR, FSYR e algoritmo de Yoon no melhor cenário de parâmetros de treinamento.

Além disso, de acordo com os resultados do teste de Wilcoxon, é preferível usar FSGR com base em modelos de MLP em vez de com base em modelos de ELM. No entanto, como não há diferença significativa entre a estabilidade dos algoritmos propostos com base nos modelos MLP e ELM quando confrontados com todas as combinações dos WBFS e dos modelos de RNA, o algoritmo ELM pode ser usado sem perder a estabilidade empregando FSGR ou FSYR. O algoritmo proposto FSOR diferiu significativamente de FSGR e FSYR ao analisar o número de posições votadas, sendo preferível utilizar FSGR ou FSYR quando utilizar a matriz de pesos obtida de modelos ELM.

5.3.5 Análise do comportamento dos WBFS em diferentes quantidades de RNAs

Analisamos os efeitos do tamanho do conjunto de RNAs usado para calcular a importância das variáveis por um WBFS. Primeiro, calculamos as medidas de estabilidade para os melhores cenários de parâmetros de cada WBFS com base em 199 conjuntos de i RNAs ($i \in \{2, 3, \cdots, 200\}$). O número 199 foi definido com base no fato de que treinamos 200 modelos de RNA para cada cenário de parâmetros, não sendo possível calcular as medidas de estabilidade com base em apenas um modelo de RNA.

Os resultados são mostrados nas Figuras 5.12, 5.13, 5.14 e 5.15. A primeira vista, somos persuadidos a concluir que pequenos conjuntos de modelos de RNA alcançam valores de confiança mais altos e uma quantidade menor de posições votadas devido ao fato de que os valores de confiança começarem altos e depois decaírem, enquanto o número de posições votadas começa baixo e em seguida aumenta.

No entanto, considerando a abordagem de votação usada para calcular as medidas de estabilidade, podemos concluir que os resultados obtidos a partir de um pequeno conjunto de modelos de RNA não podem ser confiáveis. Considere um conjunto de dados hipotético com 30 variáveis e cinco modelos de RNA usados para calcular o WBFS, em que cada ranking de importância aponta uma determinada variável para diferentes posições de importância. Ao analisar a quantidade de posições votadas ($\varepsilon = 5$), o valor de confiança mínimo ($\beta = 1/30$) e o valor de confiança ($\mathit{confiança} = 1/5$) para este caso, somos persuadidos a dizer que este é um excelente resultado. No entanto, este caso hipotético nunca teve a chance de avaliar um número considerável de RNAs diferentes para obter um resultado global.

Pelas Figuras 5.12 e 5.14 podemos notar que na maioria dos casos, no início do eixo x ($x = 2$), os valores de confiança são altos e diminuem conforme o número de

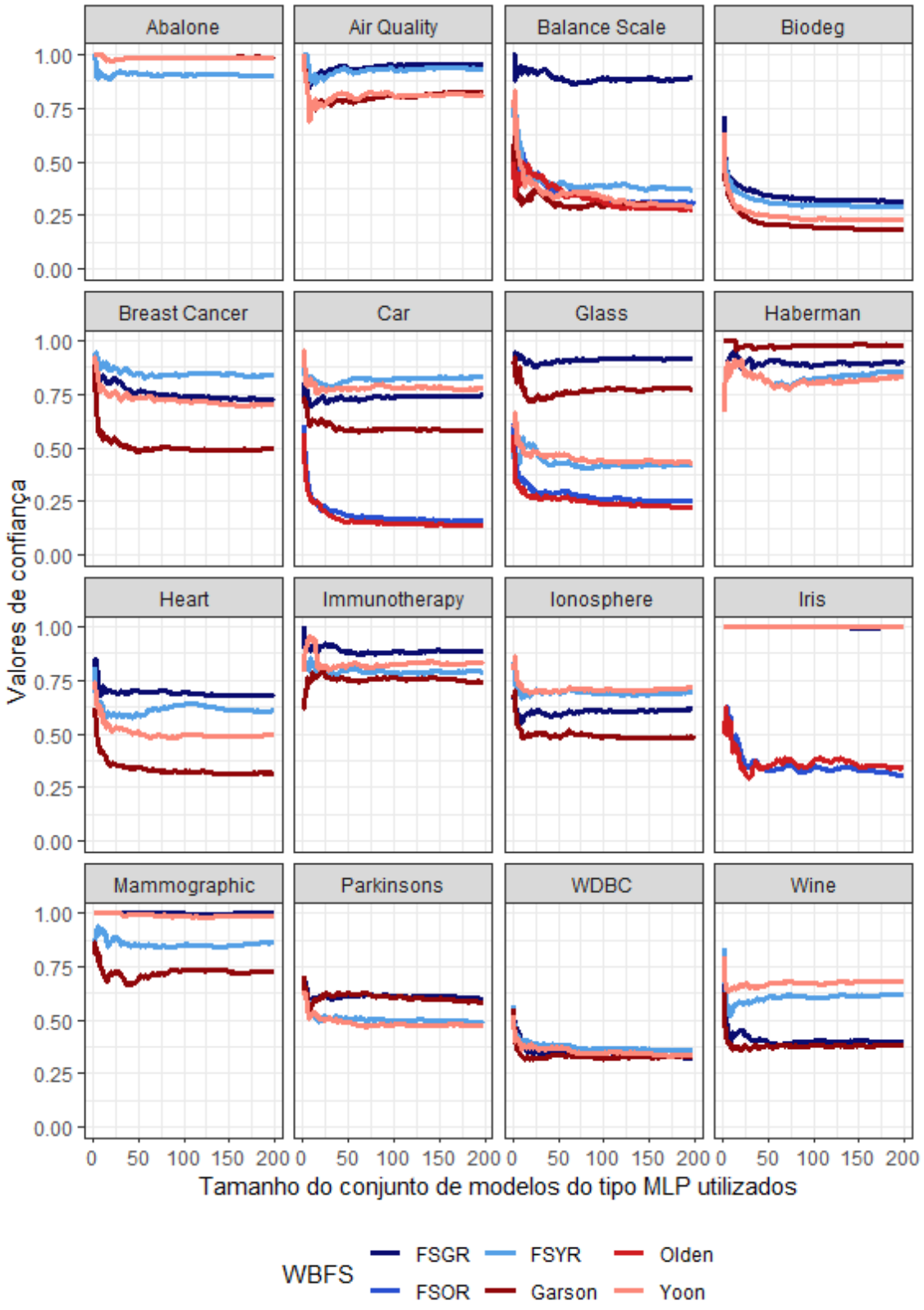


Figura 5.12: Curva de convergência dos valores de confiança no melhor cenário de parâmetro de treinamento dos modelos baseados em MLP.

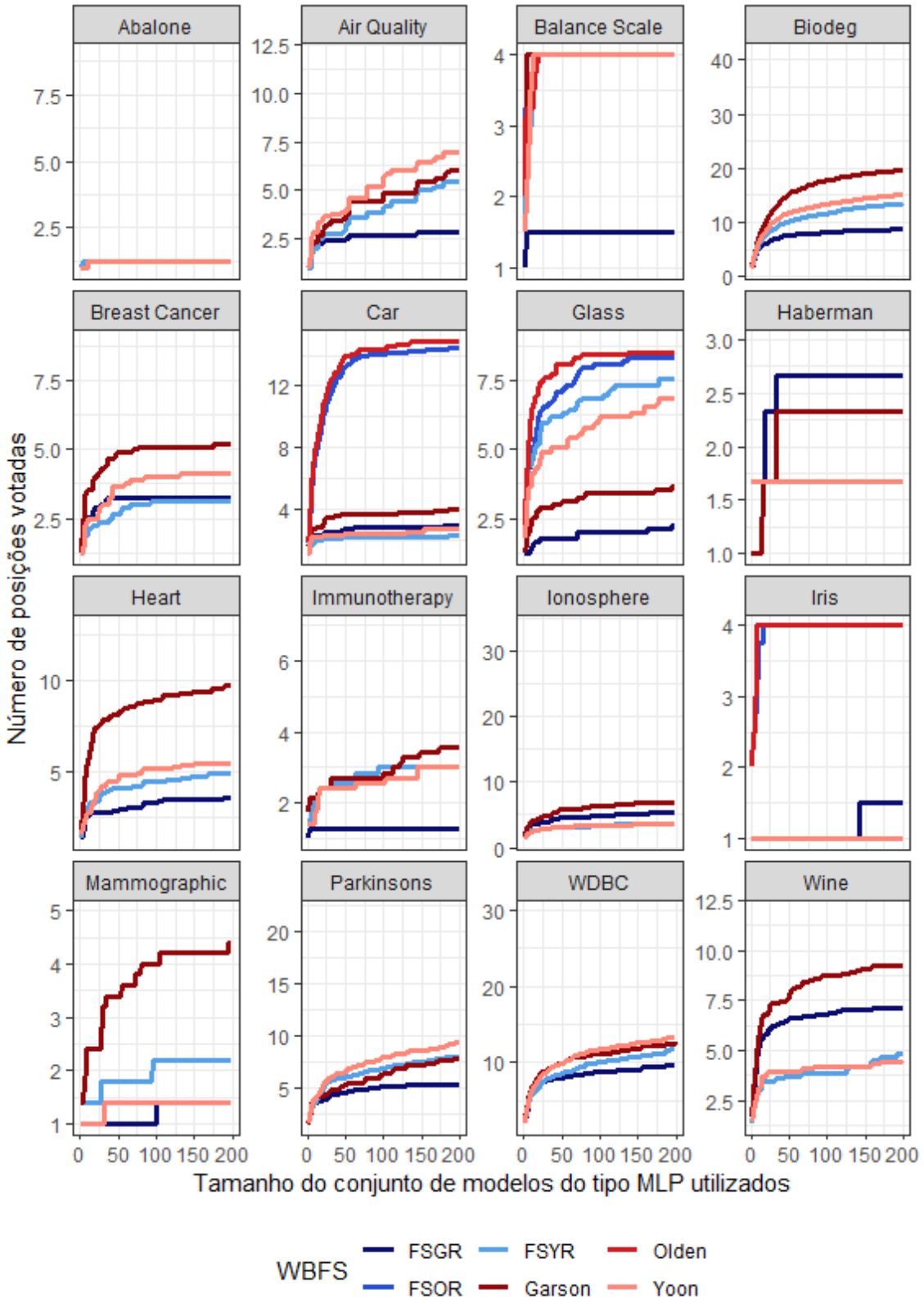


Figura 5.13: Curva de convergência do número de posições votados no melhor cenário de parâmetro de treinamento dos modelos baseados em MLP.

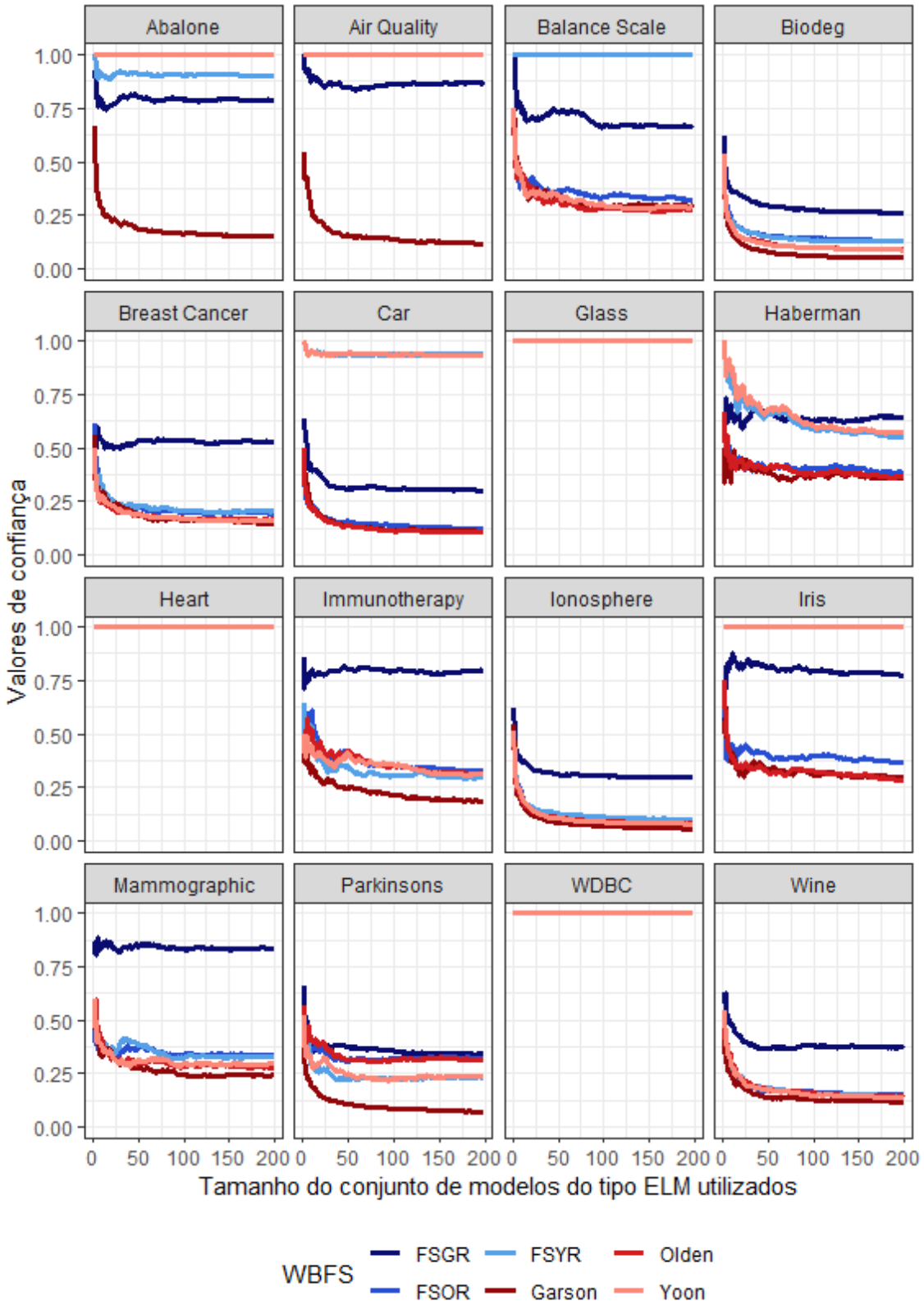


Figura 5.14: Curva de convergência dos valores de confiança no melhor cenário de parâmetro de treinamento dos modelos baseados em ELM.

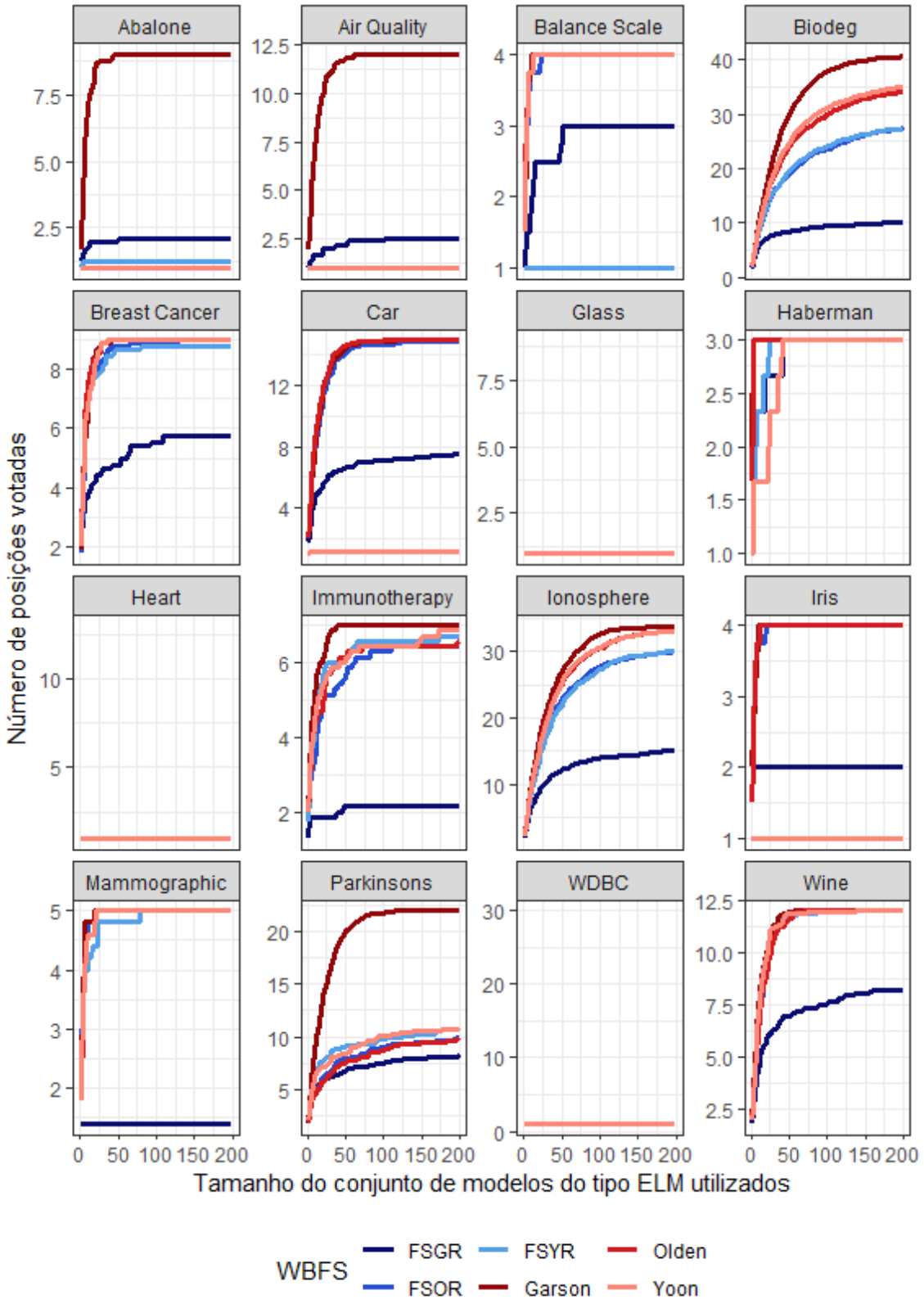


Figura 5.15: Curva de convergência do número de posições votados no melhor cenário de parâmetro de treinamento dos modelos baseados em ELM.

modelos de RNA aumenta até que x atinja um valor por volta de 25 a 40 modelos de RNA, e se estabiliza a partir de então.

Um processo semelhante ocorre nas Figuras 5.13 e 5.15. A quantidade de posições votadas começa com valores próximos a um e aumenta até que x atinja um valor em torno de 50 - 100 modelos de RNA, e se estabiliza a partir daí. Em alguns bancos de dados, a quantidade de posições votadas se estabiliza com um conjunto de RNA composto de apenas alguns modelos, e em outros casos, a quantidade de posições votadas se estabiliza apenas com um grande número de modelos de RNA ($x > 150$).

5.3.6 Estabilidade dos rankings finais de importância

Nesta subseção, examinaremos a estabilidade dos rankings de importância ao longo de três aspectos:

- O comportamento da estabilidade dos rankings individuais usadas para calcular o ranking de importância final;
- O comportamento da estabilidade dos rankings considerando os rankings finais de importância geradas a partir de cada um dos 199 tamanhos de conjuntos de RNA da subseção anterior; e,
- A estabilidade dos rankings finais de importância geradas para cada um dos WBFS no melhor cenário de parâmetros de treinamento.

A Tabela 5.7 e a Tabela 5.8 apresentam o número de diferentes rankings de importância individuais (#RI) gerados pelos 200 modelos de RNA para cada WBFS e o número de rankings finais (#RF) diferentes que foram gerados com base na combinação de modelos de RNA $i \in \{2,3,\cdots,200\}$. O número #RI representa o número de rankings individuais diferentes gerados pelo WBFS, enquanto #RF representa o número de diferentes rankings finais gerados com base na abordagem de votação em cada WBFS.

O número de #RI e #RF são altos em alguns bancos de dados. Uma forte correlação entre o número de rankings gerados e o número de variáveis no conjunto de dados foi encontrada para os rankings finais dos algoritmos de Garson, Yoon, FSGR e FSYR (Tabela 5.7). Quanto maior o número de variáveis no conjunto de dados, mais provável será de esses WBFS resultarem em um número maior de diferentes rankings de importância finais. Essa correlação não ocorre fortemente nos resultados baseados em ELM, exceto por uma forte correlação entre o número de rankings gerados e o número de variáveis na base de dados para os rankings finais dos algoritmos FSGR e FSYR (Tabela 5.8).

O teste de Friedman foi aplicado nos resultados das Tabelas 5.7 e 5.8 para identificar se existe um WBFS que resultou no menor número de rankings diferentes. Não

Tabela 5.7: Número de diferentes rankings de importância gerados pelos WBFS nos modelos do tipo MLP.

	Garson		Olden		Yoon		FSGR		FSOR		FSYR	
Conjunto de dados	#RF	#RI	#RF	#RI	#RF	#RI	#RF	#RI	#RF	#RI	#RF	#RI
Abalone	1	2	1	2	1	2	2	1	2	1	2	1
Air Quality	3	38	3	36	3	36	1	15	1	25	1	25
Balance	16	24	11	24	6	24	1	2	10	24	4	24
Biodeg	199	200	199	200	199	200	199	200	195	200	199	200
Breast cancer	14	111	2	37	2	37	1	38	1	25	1	25
Car	20	146	185	200	3	58	12	81	183	200	9	28
Glass	2	35	87	195	17	134	1	9	89	193	10	151
Haberman	1	3	2	2	2	2	1	4	2	2	2	2
Heart	74	199	15	188	15	188	8	90	7	130	7	130
Immunotherapy	4	22	1	14	1	14	1	2	2	15	2	15
Ionosphere	112	135	36	90	36	90	31	113	25	107	25	107
Iris	1	1	9	11	1	1	1	2	11	12	1	1
Mammographic	2	19	1	2	1	2	1	2	2	4	2	4
Parkinsons	15	197	73	199	73	199	33	199	38	195	38	195
WDBC	163	200	88	200	88	200	176	200	91	200	91	200
Wine	54	199	13	106	13	106	44	198	12	109	12	109
Correlação entre o número de rankings e o número de variáveis do conjunto de dados	0,89	0,68	0,53	0,56	0,86	0,73	0,81	0,77	0,48	0,61	0,82	0,78
Ranking médio dos rankings finais (χ^2 = 10,36, p valor = 0,06544) CD = 1,92	4,25		4,03		3,38		2,62		3,75		3	
Ranking médio dos rankings individuais (χ^2 = 11,98, p valor = 0,03495) CD = 11,62		4,62		3,75		3,16		3,06		3,5		2,91

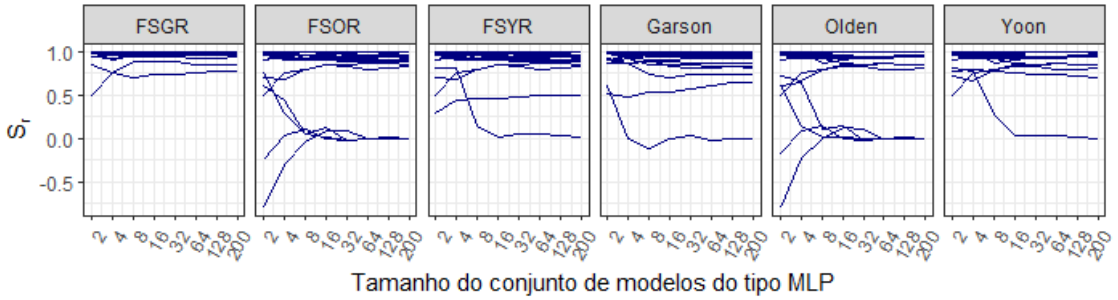
Tabela 5.8: Número de diferentes rankings de importância gerados pelos WBFS nos modelos do tipo ELM.

	Garson		Olden		Yoon		FSGR		FSOR		FSYR	
Conjunto de dados	#RF	#RI	#RF	#RI	#RF	#IR	#RF	#RI	#RF	#RI	#RF	#RI
Abalone	116	200	1	1	1	1	3	8	2	1	2	1
Air Quality	151	200	1	1	1	1	1	28	1	1	1	1
Balance	16	24	14	24	20	24	2	8	14	24	1	1
Biodeg	199	200	199	200	199	200	199	200	199	200	199	200
Breast cancer	131	200	119	200	127	200	20	133	109	200	91	199
Car	190	153	199	153	2	2	70	153	195	153	2	2
Glass	1	1	1	1	1	1	1	1	1	1	1	1
Haberman	5	6	3	6	1	6	2	6	4	6	1	6
Heart	1	1	1	1	1	1	1	1	1	1	1	1
Immunotherapy	69	134	22	102	27	86	3	9	21	93	26	76
Ionosphere	199	200	199	200	199	200	199	200	199	200	199	200
Iris	11	24	10	24	1	1	2	4	11	22	1	1
Mammographic	28	97	16	89	28	79	2	2	17	62	26	34
Parkinsons	199	200	126	177	171	192	137	200	139	180	182	185
WDBC	1	1	1	1	1	1	1	1	1	1	1	1
Wine	182	200	185	200	192	200	39	200	137	200	167	200
Correlação entre o número de rankings e o número de variáveis do conjunto de dados	0,46	0,31	0,51	0,42	0,42	0,52	0,78	0,55	0,54	0,44	0,68	0,55
Ranking médio dos rankings finais (χ^2 = 10,36, p valor = 0,06544) CD = 1,92	4,84		3,41		3,62		2,66		3,5		2,97	
Ranking médio dos rankings individuais (χ^2 = 11,98, p valor = 0,03495) CD = 11,62		4,56		3,72		3,28		3,28		3,5		2,66

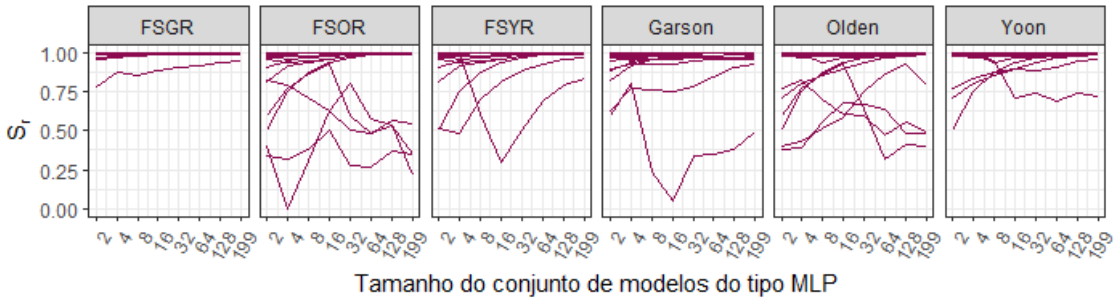
foi encontrada diferença significativa entre a quantidade de rankings finais gerados com base nos modelos de MLP ($p > 0,05$). Uma diferença significativa foi encontrada entre o número de rankings individuais gerados para ambos os modelos de RNA considerando a quantidade de rankings finais gerados com base em modelos ELM. As Tabelas 5.7 e 5.8 também apresentam os rankings médios e os resultados do teste de Friedman e os resultados do teste *post hoc* de Nemenyi. Apesar do valor de $p < 0,05$ no teste de Friedman para os rankings individuais gerados com base nos modelos MLP e ELM, o teste *post hoc* de Nemenyi não encontrou diferenças significativas entre os algoritmos (o valor da diferença crítica é maior do que as diferenças entre os rankings médios obtidos no teste de Friedman).

O número total de possíveis permutações de rankings diferentes em um conjunto de dados com n variáveis é $n!$. Dessa forma, os números mais altos apresentados nas Tabelas 5.7 e 5.8 representam uma porcentagem muito baixa de todas as permutações possíveis de rankings de importância. Por exemplo, o número de diferentes rankings de importância para os conjuntos de dados *Biodegradation* e *Ionosphere* representa cerca de 5,9e-46% e 2,2e-33% das permutações possíveis de rankings, respectivamente. Essa é uma porcentagem muito baixa. Dessa forma, calculamos a similaridade entre os rankings de importância para chegar a conclusões mais precisas.

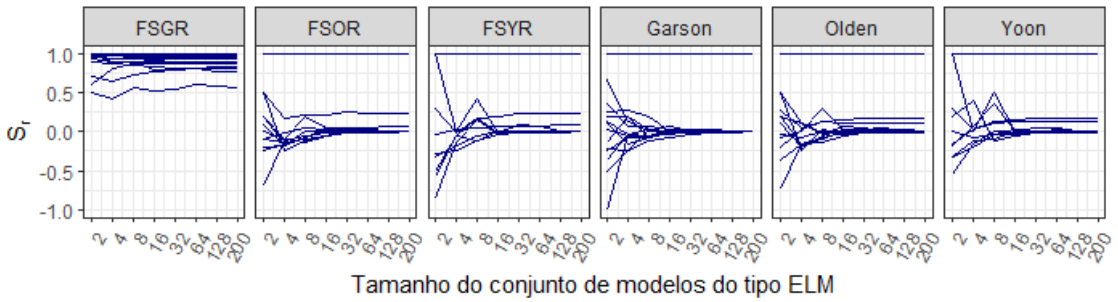
A Figura 5.16 apresenta os valores S_r para i rankings individuais (sub figuras "a)" e "c)") e para i rankings finais de importância (sub figuras "b)" e "d)"). Os conjuntos de rankings individuais foram baseados nos 200 modelos de RNA gerados a partir do melhor par de parâmetros e compostos por $i \in \{2,4,8,16,32,64,128,200\}$ posições de importância **p**. Os subconjuntos de rankings de importância finais foram compostos por $i \in \{2,4,8,16,32,64,128,199\}$ rankings de importância finais.



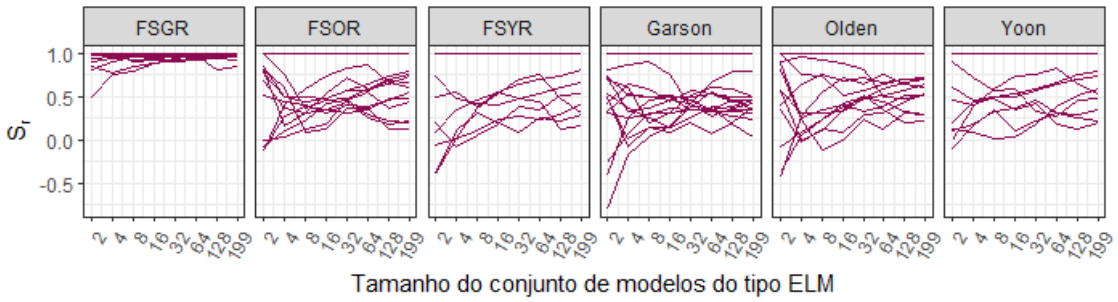
(a) Rankings individuais dos modelos baseados em MLP



(b) Rankings finais dos modelos baseados em MLP



(c) Rankings individuais dos modelos baseados em ELM



(d) Rankings finais dos modelos baseados em ELM

Figura 5.16: Resultado do teste de Nemenyi considerando os WBFS para os dois tipos de RNA estudadas.

Cada linha da Figura 5.16 representa um dos conjuntos de dados estudado. É possível observar que os rankings de importância finais sofrem uma variação maior do que os rankings de importância individuais em ambos os resultados baseados nos modelos MLP e ELM. O algoritmo FSGR, que resultou nas melhores medidas de estabilidade da abordagem de votação, também resultou nas maiores semelhanças entre os rankings de importância individual e final (valores de S_r próximos de 1). A estabilização dos valores de S_r ocorre em torno de 16 - 40 conjuntos de RNA, seguindo o padrão de estabilização dos valores de confiança apresentada na subseção anterior.

Por último, comparamos as semelhanças entre os rankings finais de importância obtidas pelos 200 modelos de RNAs em cada WBFS nos melhores cenários de parâmetros de treinamento. A Tabela 5.9 resume os resultados. A semelhança entre os rankings finais dos seis WBFS estudados em cada conjunto de dados é baixa na maioria dos casos para os modelos MLP e ELM (coluna "Comparação entre os WBFS"). Esse resultado indicou que os rankings finais de importância são bastante diferentes a depender do WBFS usado, principalmente nos rankings gerados por diferentes WBFS baseados em ELM.

Tabela 5.9: Correlação de Spearman para os rankings finais de importância

Conjunto de dados	Comparação entre os WBFS		Comparação dos tipos de RNA					
	MLP	ELM	Garson	Olden	Yoon	FSGR	FSOR	FSYR
Abalone	**0,87**	0,41	0,1	0,17	0,17	**0,97**	0,37	0,37
Air Quality	0,55	0,18	-0,55	0,17	0,17	**0,87**	0,51	0,51
Balance scale	-0,07	-0,08	-0,8	-0,4	-0,8	**1**	0,2	**0,8**
Biodegradetion	0,37	0,39	-0,08	0,3	0,13	**0,92**	0,4	0,15
Breast Cancer	0,77	0,16	0,05	0,58	0,12	**0,97**	0,22	0,12
Car	0,05	0,12	0,39	-0,64	**0,93**	**0,86**	0,3	0,77
Glass	0,12	-0,14	-0,45	-0,47	-0,12	0,68	-0,27	-0,07
Haberman	0,43	0,1	-0,5	**1**	0,5	-0,5	**1**	0,5
Heart	0,6	0,18	0,14	-0,49	-0,43	**0,93**	-0,49	-0,38
Immunotherapy	-0,33	0,18	0,32	-0,54	-0,64	**0,89**	-0,18	-0,39
Ionosphere	0,5	0,22	0,02	0,24	0,38	0,65	0,26	0,43
Iris	-0,16	-0,16	**0,8**	-0,8	0	0,6	0,6	0
Mammographic	0,05	0,09	-0,5	-0,5	0,5	**0,9**	-0,3	-0,3
Parkinson	0,72	0,16	-0,12	0	-0,24	0,7	-0,18	-0,06
WDBC	0,42	0,52	-0,13	0,02	0,04	0,71	0,06	0,06
Wine	0,16	0,19	-0,12	-0,21	0,53	0,75	-0,11	0,13

A Tabela 5.9 também apresenta a correlação entre os rankings de importância

finais obtidos por um algoritmo WBFS baseado em modelos MLP e ELM (coluna "Comparação dos tipos de RNA"). Os resultados indicam uma correlação ligeiramente maior do que os obtidos na comparação de diferentes WBFS. Em 56,25% dos conjuntos de dados foi obtida uma correlação muito forte entre os rankings finais do FSGR dos modelos MLP e ELM. Este resultado corrobora com os resultados obtidos na análise das medidas de estabilidade do WBFS com base nos modelos MLP e ELM, indicando que além de obter os maiores valores de confiança e os menores valores ε, o algoritmo FSGR é capaz de gerar rankings semelhantes quando aplicado em modelos do tipo MLP e do tipo ELM.

Com base nos resultados do teste de Nemenyi apresentados na seção 5.3.4, descobrimos que o FSGR pode ser usado com base em modelos ELM em vez de modelos MLP sem perder estabilidade significativa, além de ganhar com o menor custo computacional. Os resultados apresentados na Tabela 5.9 mostram que na maioria dos casos os rankings de importância finais do FSGR baseadas nos modelos MLP e ELM tem uma forte correlação, indicando que os rankings são parecidos. Portanto, é provável que a capacidade de predição de ambos rankings de importância também sejam fortemente correlacionados. Mais pesquisas são necessárias para investigar a capacidade de predição de subconjuntos de variáveis criadas a partir dos rankings de importância finais obtidos a partir de diferentes modelos de RNA.

5.4 Discussão

Os resultados apresentados neste capítulo ilustram que todos os três algoritmos propostos, FSGR, FSOR e FSYR podem aumentar a estabilidade dos rankings. Os algoritmos propostos usam o seletor de variáveis do tipo filtro ReliefF junto com os algoritmos de Garson, Olden e Yoon.

Com base nos resultados apresentados neste estudo, é altamente recomendável usar os resultados do WBFS como uma ferramenta de pré-processamento em vez de uma ferramenta de inferência. Em estudos recentes de diferentes áreas de pesquisa, os algoritmos WBFS têm sido usados como uma ferramenta de inferência para fazer suposições sobre a importância das variáveis para melhor compreender o processo subjacente, majoritariamente baseados em apenas um modelo de RNA. No entanto, como demonstrado neste estudo, os rankings finais de importância podem variar dependendo do conjunto de RNA usado para calcular o WBFS, e como este algoritmo usa a matriz de pesos de RNA, os resultados podem não ser reproduzíveis devido à inicialização aleatória e ao treinamento em si.

Ao usar o WBFS como ferramenta de pré-processamento, as variáveis podem ser usadas como *forward selection* ou *backward elimination* como entrada para um

classificador, por exemplo, caracterizando um seletor de variáveis do tipo *wrapper*. Nessa abordagem, a importância das variáveis será validada pelo desempenho do classificador. Considerando a baixa estabilidade do WBFS, o desempenho da classificação é uma métrica que pode ser usada para validar a importância das variáveis.

Além disso, nosso estudo fornece uma contribuição teórica a respeito da relação entre os modelos de RNA e a influência de seus parâmetros de treinamento nas estabilidades de WBFS. Valores de *weight decay* mais altos podem resultar em melhores medidas de estabilidades com base nos modelos MLP, e a função de ativação linear também pode resultar em melhores medidas de estabilidades com base nos modelos ELM. Esperamos que novas pesquisas possam propor novos métodos de WBFS que possam fornecer resultados estáveis para serem usados como um pré-processamento e como uma ferramenta de inferência com resultados reproduzíveis.

Reconhecemos que este estudo está focado em um espaço de parâmetros específicos de apenas modelos MLP e ELM. Existem outros parâmetros e outros modelos de RNA que podem afetar a estabilidade do WBFS. Apesar desta limitação, este estudo ainda fornece informações úteis que podem ser usadas para melhorar a aplicabilidade do WBFS no futuro.

Um WBFS pode resultar em diferentes rankings de importância final. Esses rankings alcançam uma similaridade estável com o uso de aproximadamente 30 modelos de RNA, nos quais os melhores resultados foram obtidos a partir do algoritmo FSGR. Os rankings finais de importância tendem a ser diferentes dependendo do WBFS empregado. Assim, cada um dos rankings finais pode fornecer informações diferentes. No entanto, a robustez da capacidade de predição dos subconjuntos de variáveis gerados com base nos algoritmos WBFS está além do escopo deste estudo.

O ponto forte do estudo apresentado neste capítulo é que ele analisa as medidas de estabilidade em função de diferentes modelos de RNA e parâmetros. Com base nos resultados obtidos na análise dos parâmetros foi possível analisar a individualidade de WBFS. Foi observado que as melhorias propostas nos algoritmos WBFS forneceram um resultado adequado em comparação com os algoritmos originais e não modificados (algoritmos de Garson, Olden e Yoon). Este estudo também mostra o número de modelos de RNA que um WBFS precisa para fornecer resultados estáveis, na faixa de estabilidade que cada WBFS pode atingir. Apesar disso, mais pesquisas são necessárias para melhorar a estabilidade dos algoritmos WBFS.

Ademais, demonstramos que o uso de ELM para gerar a matriz de peso para calcular o WBFS pode fornecer rankings estáveis equivalentes em comparação com os modelos MLP ao utilizar o algoritmo FSGR. Pesquisas futuras podem avaliar o uso de ELM e MLP em conjuntos de dados de alta dimensionalidade e usar os resultados sobre os melhores parâmetros apresentados neste estudo como ponto de partida.

5.5 Considerações finais

Com o objetivo de melhorar a estabilidade do WBFS, apresentamos três novos algoritmos chamados FSGR, FSOR e FSYR, que são uma melhoria dos algoritmos clássicos propostos pelos algoritmos de Garson, Olden e Yoon, combinando esses algoritmos com a técnica ReliefF. Os resultados da comparação sugerem fortemente que uma de nossas propostas de algoritmo, o FSGR, é mais eficaz do que os outros algoritmos. No entanto, mais pesquisas são necessárias para melhorar os WBFS a ponto de se tornarem mais estáveis a ponto de usar esses algoritmos como uma ferramenta de inferência.

Os seis algoritmos foram testados com base em diferentes cenários de parâmetros de modelos MLP e ELM. Foi encontrada uma diferença significativa entre os diferentes valores de *weight decay* no modelo MLP, indicando que maiores valores de *weight decay* resultam em maior estabilidade. As diferentes funções de ativação nos modelos ELM também resultaram em uma diferença significativa, em que a função de ativação linear obteve resultados superiores.

Para confirmar a estabilidade dos rankings de importância finais dos algoritmos propostos, investigamos as diferenças entre os rankings de importância finais em relação ao número de modelos de RNA usados para gerá-la. Demonstramos que os algoritmos WBFS alcançam resultados estáveis usando a abordagem de votação em cerca de 25-40 modelos de RNAs. Os rankings finais de importância podem ser diferentes dependendo do conjunto de RNAs. É importante utilizar o ranking de importância final com a validação de outros seletores de variáveis, ou mesmo a validação de um modelo de classificação, que pode medir a capacidade de predição das variáveis selecionadas.

A validação experimental das melhorias propostas de WBFS e a extensa comparação de WBFS em diferentes cenários mostra a adequação dos algoritmos propostos e dá uma visão sobre os benefícios e limitações dos algoritmos de WBFS.

CAPÍTULO 6

Conclusões e trabalhos futuros

Este estudo apresenta a teoria por trás dos seletores de variáveis baseados em redes neurais artificiais, uma revisão de literatura das formas de uso dos WBFS, a proposta de uma nova abordagem para medir a estabilidade dos WBFS, a proposta de melhoria dos algoritmos existentes e um estudo sobre a influência de diferentes tipos de redes neurais artificiais e parâmetros de treinamento na estabilidade de um WBFS.

O objetivo central deste trabalho foi avaliar as condições de uso dos WBFS e melhorar sua aplicabilidade em pesquisas científicas. Apesar de esses algoritmos serem bastante utilizados em diferentes áreas de pequisa, até o presente estudo não havia pesquisas que detalhavam a forma de uso destes algoritmos, e poucos estudos tentaram lidar com a instabilidade destes métodos por meio da média dos valores de importância.

A primeira contribuição deste trabalho se deu com a definição e quantificação das formas de uso dos WBFS por meio de três categorias. A primeira categoria diz respeito a utilizar um WBFS com base em apenas uma rede neural; a segunda categoria diz respeito a utilizar um WBFS com base em uma rede neural com o melhor desempenho entre um conjunto de redes neurais; e por fim, a terceira categoria diz respeito a utilizar a média dos valores de importância obtidos pela aplicação de um WBFS em um conjunto de redes neurais artificiais.

Intrigados pelo uso da média dos valores de importância e o potencial problema de que uma variável poderia ser classificada em praticamente todas as posições de importância possíveis, desenvolvemos uma nova abordagem para medir a instabilidade dos WBFS, caracterizando a segunda contribuição deste estudo e a quarta categoria de uso dos WBFS.

A partir da proposta da abordagem de votação foi possível concluir que o uso da média dos valores de importância, assim como o uso dos WBFS em apenas um único modelo de RNA pode levar a resultados errôneos devido às diferenças obtidas nos rankings de importância em decorrência da inicialização e treinamento dos pesos das redes neurais.

A terceira contribuição desta tese estabelece um ponto de partida na relação entre o tipo e os parâmetros de redes neurais que podem influenciar a estabilidade dos

WBFS. Apesar de os experimentos abrangerem poucos parâmetros e apenas dois tipos de redes neurais, os resultados demonstram diferença significativa entre as estabilidades dos WBFS a depender do tipo e dos parâmetros de treinamento utilizados nas redes neurais.

Por fim, a quarta contribuição deste trabalho se baseia na proposta de melhoria dos WBFS por meio da combinação dos algoritmos com o seletor de variáveis ReliefF. O novo algoritmo baseado no algoritmo de Garson e no seletor de variáveis ReliefF (FSGR) demonstrou obter os resultados mais estáveis, tanto no uso de MLP como no uso de ELM, superando os algoritmos de Garson, Olden, Yoon, e os novos algoritmos FSOR e FSYR.

Ainda há o que ser explorado sobre os temas aqui estudados. Os experimentos realizados nesta tese focaram nos valores de estabilidade dos WBFS mediante diferentes cenários, sendo possível ainda, estimar e comparar a capacidade de predição dos rankings finais de importância das melhorias dos algoritmos propostos. Além disso, ainda há outros tipos de redes neurais e parâmetros de treinamento que podem ser avaliados em relação à estabilidade dos WBFS, como as redes neurais recorrentes, ELM com regularização e diferentes funções de ativação em conjunto com MLP com a regularização de *weight decay*.

Ressaltamos que os resultados apresentados nesta tese são um ponto de partida para diversas pesquisas devido ao fato de este ser o primeiro trabalho a investigar a estabilidade dos WBFS. Esperamos que no futuro possam existir WBFS com um nível de estabilidade aceitável, para assim, estimar a importância das variáveis de entrada de um modelo de RNA sem a necessidade de validação por outro classificador.

Referências Bibliográficas

[1] J. Abellán García, J. Fernández Gómez, and N. Torres Castellanos. Properties prediction of environmentally friendly ultra-high-performance concrete using artificial neural networks. *European Journal of Environmental and Civil Engineering*, pages 1–25, 2020.

[2] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018.

[3] R. Acharyya. Finite element investigation and ann-based prediction of the bearing capacity of strip footings resting on sloping ground. *International Journal of Geo-Engineering*, 10(1):5, 2019.

[4] R. Acharyya, A. Dey, and B. Kumar. Finite element and ann-based prediction of bearing capacity of square footing resting on the crest of c-φ soil slope. *International Journal of Geotechnical Engineering*, 2018.

[5] A. Alnedawi, K. P. Nepal, and R. Al-Ameri. Effect of loading frequencies on permanent deformation of unbound granular materials. *International Journal of Pavement Engineering*, pages 1–9, 2019.

[6] B. Amiri, R. Dizène, and K. Dahmani. Most relevant input parameters selection for 10-min global solar irradiation estimation on arbitrary inclined plane using neural networks. *International Journal of Sustainable Energy*, pages 1–25, 2020.

[7] E. M. Araújo, M. D. de Lima, R. Barbosa, and L. R. F. Alleoni. Using machine learning and multi-element analysis to evaluate the authenticity of organic and conventional vegetables. *Food Analytical Methods*, 12(11):2542–2554, 2019.

[8] N. Arora and P. D. Kaur. A bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment. *Applied Soft Computing*, 86:105936, 2020.

[9] M. W. Beck. Neuralnettools: visualization and analysis tools for neural networks. *Journal of statistical software*, 85(11):1, 2018.

[10] N. V. Bhattacharjee and E. W. Tollner. Improving management of windrow composting systems by modeling runoff water quality dynamics using recurrent neural network. *Ecological Modelling*, 339:68–76, 2016.

[11] N. Bhuyan, R. Narzari, S. M. B. Baruah, and R. Kataki. Comparative assessment of artificial neural network and response surface methodology for evaluation of the predictive capability on bio-oil yield of tithonia diversifolia pyrolysis. *Biomass Conversion and Biorefinery*, pages 1–16, 2020.

[12] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[13] X.-N. Bui, H. Nguyen, H.-A. Le, H.-B. Bui, and N.-H. Do. Prediction of blast-induced air over-pressure in open-pit mine: assessment of different artificial intelligence techniques. *Natural Resources Research*, 29(2):571–591, 2020.

[14] J. Cai, J. Luo, S. Wang, and S. Yang. Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79, 2018.

[15] A. Chakraborty, D. Mukherjee, and S. Mitra. Development of pedestrian crash prediction model for a developing country using artificial neural network. *International journal of injury control and safety promotion*, 26(3):283–293, 2019.

[16] G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[17] N. L. Costa, L. A. G. Llobodanin, I. A. Castro, and R. Barbosa. Geographical classification of tannat wines based on support vector machines and feature selection. *Beverages*, 4(4):97, 2018.

[18] N. L. Costa, L. A. G. Llobodanin, I. A. Castro, and R. Barbosa. Finding the most important sensory descriptors to differentiate some vitis vinifera l. south american wines using support vector machines. *European Food Research and Technology*, 245(6):1207–1228, 2019.

[19] N. L. Costa, L. A. G. Llobodanin, I. A. Castro, and R. Barbosa. Using support vector machines and neural networks to classify merlot wines from south america. *Information Processing in Agriculture*, 6(2):265–278, 2019.

[20] N. L. da Costa, M. S. da Costa, and R. Barbosa. A review on the application of chemometrics and machine learning algorithms to evaluate beer authentication. *Food Analytical Methods*, pages 1–20, 2020.

[21] N. L. da Costa, M. D. de Lima, and R. Barbosa. Evaluation of feature selection methods based on artificial neural network weights. *Expert Systems with Applications*, page 114312, 2020.

[22] N. L. da Costa, L. A. G. Llobodanin, I. A. Castro, and R. Barbosa. The use of data mining to classify carménère and merlot wines from chile. *Expert Systems*, 36(2):e12361, 2019.

[23] N. L. da Costa, L. A. G. Llobodanin, M. D. de Lima, I. A. Castro, and R. Barbosa. Geographical recognition of syrah wines by combining feature selection with extreme learning machine. *Measurement*, 120:92–99, 2018.

[24] N. L. da Costa, J. P. B. Ximenez, J. L. Rodrigues, F. Barbosa, and R. Barbosa. Characterization of cabernet sauvignon wines from california: determination of origin based on icp-ms analysis and machine learning techniques. *European Food Research and Technology*, pages 1–13, 2020.

[25] M. Dastkhoon, M. Ghaedi, A. Asfaram, M. H. A. Azqhandi, and M. K. Purkait. Simultaneous removal of dyes onto nanowires adsorbent use of ultrasound assisted adsorption to clean waste water: chemometrics for modeling and optimization, multicomponent adsorption and kinetic study. *Chemical Engineering Research and Design*, 124:222–237, 2017.

[26] M. D. de Lima, N. L. Costa, and R. Barbosa. Improvements on least squares twin multi-class classification support vector machine. *Neurocomputing*, 313:196–205, 2018.

[27] J. De Oña and C. Garrido. Extracting the contribution of independent variables in neural network models: a new approach to handle instability. *Neural Computing and Applications*, 25(3-4):859–869, 2014.

[28] A. Dehghanbanadaki, M. A. Sotoudeh, I. Golpazir, A. Keshtkarbanaeemoghadam, and M. Ilbeigi. Prediction of geotechnical properties of treated fibrous peat by artificial neural networks. *Bulletin of Engineering Geology and the Environment*, 78(3):1345–1358, 2019.

[29] B. Demirbay, D. B. Kara, and Ş. Uğur. A bayesian regularized feed-forward neural network model for conductivity prediction of ps/mwcnt nanocomposite film coatings. *Applied Soft Computing*, 96:106632, 2020.

[30] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

[31] O. Ellegaard and J. A. Wallin. The bibliometric analysis of scholarly production: How great is the impact? *Scientometrics*, 105(3):1809–1831, 2015.

[32] Y. Erzin and D. Turkoz. Use of neural networks for the prediction of the cbr value of some aegean sands. *Neural Computing and Applications*, 27(5):1415–1426, 2016.

[33] A. Fallah, A. A. Oladipo, and M. Gazi. Boron-doped sucrose carbons for supercapacitor electrode: artificial neural network-based modelling approach. *Journal of Materials Science: Materials in Electronics*, 31(17):14563–14576, 2020.

[34] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37, 1996.

[35] A. Fischer. How to determine the unique contributions of input-variables to the nonlinear regression function of a multilayer perceptron. *Ecological Modelling*, 309:60–63, 2015.

[36] D.-L. Flores, C. Gómez, D. Cervantes, A. Abaroa, C. Castro, and R. A. Castaneda-Martínez. Predicting the physiological response of tivela stultorum hearts with digoxin from cardiac parameters using artificial neural networks. *Biosystems*, 151:1–7, 2017.

[37] S. M. Formentin and B. Zanuttigh. A methodological approach for the development and verification of artificial neural networks based on an application to wave–structure interaction processes. *Coastal Engineering Journal*, 60(3):260–279, 2018.

[38] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[39] R. Garcel, O. León, and E. Magaz. Preliminary modeling of an industrial recombinant human erythropoietin purification process by artificial neural networks. *Brazilian Journal of Chemical Engineering*, 32(3):725–734, 2015.

[40] H. García-Martínez, H. Flores-Magdaleno, R. Ascencio-Hernández, A. Khalil-Gardezi, L. Tijerina-Chávez, O. R. Mancilla-Villa, and M. A. Vázquez-Peña. Corn grain yield estimation from vegetation indices, canopy cover, plant density, and a neural network using multispectral and rgb images acquired with unmanned aerial vehicles. *Agriculture*, 10(7):277, 2020.

[41] C. Garrido, R. De Oña, and J. De Oña. Neural networks for analyzing service quality in public transportation. *Expert Systems with Applications*, 41(15):6830–6838, 2014.

[42] G. D. Garson. Interpreting neural-network connection weights. *AI expert*, 6(4):46–51, 1991.

[43] M. Gevrey, I. Dimopoulos, and S. Lek. Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological modelling*, 160(3):249–264, 2003.

[44] M. A. Ghorbani, M. T. Aalami, and L. Naghipour. Use of artificial neural networks for electrical conductivity modeling in asi river. *Applied Water Science*, 7(4):1761–1772, 2017.

[45] B. Godsey. *Think like a data scientist*. Pearson Professional Computing, 2017.

[46] N. Golshani, R. Shabanpour, S. M. Mahmoudifard, S. Derrible, and A. Mohammadian. Modeling travel mode and timing decisions: Comparison of artificial neural networks and copula-based joint model. *Travel Behaviour and Society*, 10:21–32, 2018.

[47] J. C. Gomes, V. A. Barbosa, D. E. Ribeiro, R. E. de Souza, and W. P. dos Santos. Electrical impedance tomography image reconstruction based on backprojection and extreme learning machines. *Research on Biomedical Engineering*, pages 1–12, 2020.

[48] G. Gualtieri, F. Camilli, A. Cavaliere, T. De Filippis, F. Di Gennaro, S. Di Lonardo, F. Dini, B. Gioli, A. Matese, W. Nunziati, et al. An integrated low-cost road traffic and air pollution monitoring platform to assess vehicles air quality impact in urban areas. *Transportation research procedia*, 27:609–616, 2017.

[49] G. Gualtieri, F. Carotenuto, S. Finardi, M. Tartaglia, P. Toscano, and B. Gioli. Forecasting pm10 hourly concentrations in northern italy: Insights on models performance and pm10 drivers through self-organizing maps. *Atmospheric Pollution Research*, 9(6):1204–1213, 2018.

[50] L. Guo, C. J. Vargo, Z. Pan, W. Ding, and P. Ishwar. Big social data analytics in journalism and mass communication: Comparing dictionary-based text analysis and unsupervised topic modeling. *Journalism & Mass Communication Quarterly*, 93(2):332–359, 2016.

[51] T. V. Ha, T. Asada, and M. Arimura. Determination of the influence factors on household vehicle ownership patterns in phnom penh using statistical and machine learning methods. *Journal of Transport Geography*, 78:70–86, 2019.

[52] A. M. Hashem, M. E. M. Rasmy, K. M. Wahba, and O. G. Shaker. Single stage and multistage classification models for the prediction of liver fibrosis degree in patients with chronic hepatitis c infection. *Computer methods and programs in biomedicine*, 105(3):194–209, 2012.

[53] S. S. Haykin et al. Neural networks and learning machines/simon haykin., 2009.

[54] A. A. Hedayat, A. Iranpour, and E. A. Afzadi. Evaluation of the bolt and weld eccentricities of conventional single-plate shear connections. *Journal of Constructional Steel Research*, 153:254–274, 2019.

[55] M. S. Hossain and G. Muhammad. Emotion recognition using deep learning approach from audio–visual emotional big data. *Information Fusion*, 49:69–78, 2019.

[56] M. Hosseinpour, Y. Sharifi, and H. Sharifi. Neural network application for distortional buckling capacity assessment of castellated steel beams. In *Structures*, volume 27, pages 1174–1183. Elsevier, 2020.

[57] P. Howes and N. Crook. Using input parameter influences to support the decisions of feedforward neural networks. *Neurocomputing*, 24(1-3):191–206, 1999.

[58] G.-B. Huang, D. H. Wang, and Y. Lan. Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, 2(2):107–122, 2011.

[59] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[60] A. K. Jain, R. P. Duin, and J. Mao. Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37, 2000.

[61] Y. Jbari and S. Abderafi. Parametric study to enhance performance of wastewater treatment process, by reverse osmosis-photovoltaic system. *Applied Water Science*, 10(10):1–14, 2020.

[62] I. d. S. T. Júnior, C. M. M. E. Torres, H. G. Leite, N. L. M. de Castro, C. P. B. Soares, R. V. O. Castro, and A. A. Farias. Machine learning: Modeling increment in diameter of individual trees on atlantic forest fragments. *Ecological Indicators*, 117:106685, 2020.

[63] R. M. Kakhki, M. Mohammadpoor, R. Faridi, and M. Bahadori. The development of an artificial neural network–genetic algorithm model (ann-ga) for the adsorption and photocatalysis of methylene blue on a novel sulfur–nitrogen co-doped fe 2 o 3 nanostructure surface. *RSC Advances*, 10(10):5951–5960, 2020.

[64] J. Kang, H. Wang, F. Yuan, Z. Wang, J. Huang, and T. Qiu. Prediction of precipitation based on recurrent neural networks in jingdezhen, jiangxi province, china. *Atmosphere*, 11(3):246, 2020.

[65] U. M. Khaire and R. Dhanalakshmi. Stability of feature selection algorithm: A review. *Journal of King Saud University-Computer and Information Sciences*, 2019.

[66] K. Kira, L. A. Rendell, et al. The feature selection problem: Traditional methods and a new algorithm. In *Aaai*, volume 2, pages 129–134, 1992.

[67] I. Kononenko. Estimating attributes: analysis and extensions of relief. In *European conference on machine learning*, pages 171–182. Springer, 1994.

[68] Y. Korteby, K. Kristó, T. Sovány, and G. Regdon Jr. Use of machine learning tool to elucidate and characterize the growth mechanism of an in-situ fluid bed melt granulation. *Powder Technology*, 331:286–295, 2018.

[69] S. Krishnan and Y. Athavale. Trends in biomedical signal feature extraction. *Biomedical Signal Processing and Control*, 43:41–63, 2018.

[70] A. Kumar, M. Ramachandran, A. H. Gandomi, R. Patan, S. Lukasik, and R. K. Soundarapandian. A deep neural network based classifier for brain tumor diagnosis. *Applied Soft Computing*, 82:105528, 2019.

[71] V. Kumar and A. Kumar. Predicting the settlement of raft resting on sand reinforced with planar and geocell using generalized regression neural networks (grnn) and back propagated neural networks (bpnn). *International Journal of Geosynthetics and Ground Engineering*, 4(4):30, 2018.

[72] R. J. Kwaria, E. A. Q. Mondarte, H. Tahara, R. Chang, and T. Hayashi. Data-driven prediction of protein adsorption on self-assembled monolayers toward material screening and design. *ACS Biomaterials Science & Engineering*, 6(9):4949–4956, 2020.

[73] T. Kwartler. *Text mining in practice with R*. John Wiley & Sons, 2017.

[74] J. Lee, C.-G. Kim, J. E. Lee, N. W. Kim, and H. Kim. Application of artificial neural networks to rainfall forecasting in the geum river basin, korea. *Water*, 10(10):1448, 2018.

[75] J. Lee, C.-G. Kim, J. E. Lee, N. W. Kim, and H. Kim. Medium-term rainfall forecasts using artificial neural networks with monte-carlo cross-validation and aggregation for the han river basin, korea. *Water*, 12(6):1743, 2020.

[76] S. Lee and H. Ahn. The hybrid model of neural networks and genetic algorithms for the design of controls for internet-based systems for business-to-consumer electronic commerce. *Expert Systems with Applications*, 38(4):4326–4338, 2011.

[77] S. Lee and W. S. Choi. A multi-industry bankruptcy prediction model using back-propagation neural network and multivariate discriminant analysis. *Expert Systems with Applications*, 40(8):2941–2946, 2013.

[78] T. Lei, X.-h. Sun, X.-h. Guo, and J.-j. Ma. Quantifying the relative importance of soil moisture, nitrogen, and temperature on the urea hydrolysis rate. *Soil Science and Plant Nutrition*, 63(3):225–232, 2017.

[79] W. Lin, L. Jing, Z. Zhu, Q. Cai, and B. Zhang. Removal of heavy metals from mining wastewater by micellar-enhanced ultrafiltration (meuf): experimental investigation and monte carlo-based artificial neural network modeling. *Water, Air, & Soil Pollution*, 228(6):206, 2017.

[80] J. Liu, L. N. Boyle, and A. G. Banerjee. Predicting interstate motor carrier crash rate level using classification models. *Accident Analysis & Prevention*, 120:211–218, 2018.

[81] Q. Liu, X. Cui, Y.-C. Chou, M. F. Abbod, J. Lin, and J.-S. Shieh. Ensemble artificial neural networks applied to predict the key risk factors of hip bone fracture for elders. *Biomedical Signal Processing and Control*, 21:146–156, 2015.

[82] W. Liu, Y.-L. Li, M.-T. Feng, Y.-W. Zhao, X. Ding, B. He, and X. Liu. Application of feedback system control optimization technique in combined use of dual antiplatelet therapy and herbal medicines. *Frontiers in physiology*, 9:491, 2018.

[83] Y. Mahdi and K. Daoud. Microdroplet size prediction in microfluidic systems via artificial neural network modeling for water-in-oil emulsion formulation. *Journal of Dispersion Science and Technology*, 38(10):1501–1508, 2017.

[84] B. Mak and R. W. Blanning. An empirical measure of element contribution in neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(4):561–564, 1998.

[85] E. Manzanarez-Ozuna, D.-L. Flores, E. Gutiérrez-López, D. Cervantes, and P. Juárez. Model based on ga and dnn for prediction of mrna-smad7 expression regulated by mirnas in breast cancer. *Theoretical Biology and Medical Modelling*, 15(1):24, 2018.

[86] R. May, G. Dandy, and H. Maier. Review of input variable selection methods for artificial neural networks. *Artificial neural networks-methodological advances and biomedical applications*, 10:16004, 2011.

[87] S. M. Mousavi, E. S. Mostafavi, and P. Jiao. Next generation prediction model for daily solar radiation on horizontal surface using a hybrid neural network and simulated annealing method. *Energy Conversion and Management*, 153:671–682, 2017.

[88] R. Muñoz-Mas, J. Sánchez-Hernández, M. E. McClain, R. Tamatamah, S. C. Mukama, and F. Martínez-Capel. Investigating the influence of habitat structure and hydraulics on tropical macroinvertebrate communities. *Ecohydrology & Hydrobiology*, 19(3):339–350, 2019.

[89] A. K. Nayak and A. Pal. Statistical modeling and performance evaluation of biosorptive removal of nile blue a by lignocellulosic agricultural waste under the application of high-strength dye concentrations. *Journal of Environmental Chemical Engineering*, 8(2):103677, 2020.

[90] V. Nourani and M. S. Fard. Sensitivity analysis of the artificial neural network outputs in simulation of the evaporation process at different climatologic regimes. *Advances in Engineering Software*, 47(1):127–146, 2012.

[91] J. D. Olden and D. A. Jackson. Illuminating the black box: a randomization approach for understanding variable contributions in artificial neural networks. *Ecological modelling*, 154(1-2):135–150, 2002.

[92] J. D. Olden, M. K. Joy, and R. G. Death. An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological Modelling*, 178(3-4):389–397, 2004.

[93] H. Park, A. Haghani, and X. Zhang. Interpretation of bayesian neural networks for predicting the duration of detected incidents. *Journal of Intelligent Transportation Systems*, 20(4):385–400, 2016.

[94] F. Parvizian, M. Rahimi, and S. Hosseini. Prediction of the characteristics of a new sonochemical reactor using an expert model. *Chemical Engineering Communications*, 203(5):683–691, 2016.

[95] M. Pejic-Bach, T. Bertoncel, M. Meško, and Ž. Krstić. Text mining of industry 4.0 job advertisements. *International Journal of Information Management*, 50:416–431, 2020.

[96] N. M. Peleato, R. L. Legge, and R. C. Andrews. Neural networks for dimensionality reduction of fluorescence spectra and prediction of drinking water disinfection byproducts. *Water research*, 136:84–94, 2018.

[97] K. Pentoś. The methods of extracting the contribution of variables in artificial neural network models–comparison of inherent instability. *Computers and Electronics in Agriculture*, 127:141–146, 2016.

[98] K. Pentoś and D. Łuczycka. Dielectric properties of honey: the potential usability for quality assessment. *European Food Research and Technology*, 244(5):873–880, 2018.

[99] D. Pérez-Zárate, E. Santoyo, A. Acevedo-Anicasio, L. Díaz-González, and C. García-López. Evaluation of artificial neural networks for the prediction of deep reservoir temperatures using the gas-phase composition of geothermal fluids. *Computers & Geosciences*, 129:49–68, 2019.

[100] A. Pimenta, D. Carneiro, J. Neves, and P. Novais. A neural network to classify fatigue from human–computer interaction. *Neurocomputing*, 172:413–426, 2016.

[101] S. Piramuthu. Evaluating feature selection methods for learning in data mining applications. *European journal of operational research*, 156(2):483–494, 2004.

[102] R. Pires dos Santos, D. Dean, J. Weaver, and Y. Hovanski. Identifying the relative importance of predictive variables in artificial neural networks based on data produced through a discrete event simulation of a manufacturing environment. *International Journal of Modelling and Simulation*, 39(4):234–245, 2019.

[103] T. Punshon, Z. Li, B. P. Jackson, W. T. Parks, M. Romano, D. Conway, E. R. Baker, and M. R. Karagas. Placental metal concentrations in relation to placental growth, efficiency and birth weight. *Environment international*, 126:533–542, 2019.

[104] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.

[105] F. Rahimpour, T. Shojaeimehr, and M. Sadeghi. Biosorption of pb (ii) using gundelia tournefortii: Kinetics, equilibrium, and thermodynamics. *Separation Science and Technology*, 52(4):596–607, 2017.

[106] G. S. Raman and M. S. Klima. Application of statistical and machine learning techniques for laboratory-scale pressure filtration: Modeling and analysis of cake moisture. *Mineral Processing and Extractive Metallurgy Review*, 40(2):148–155, 2019.

[107] R. Ranasinghe, M. Jaksa, Y. Kuo, and F. P. Nejad. Application of artificial neural networks for predicting the impact of rolling dynamic compaction using dynamic

cone penetrometer test results. *Journal of Rock Mechanics and Geotechnical Engineering*, 9(2):340–349, 2017.

[108] S. Rathore, M. Habes, M. A. Iftikhar, A. Shacklett, and C. Davatzikos. A review on neuroimaging-based classification studies and associated feature extraction methods for alzheimer's disease and its prodromal stages. *NeuroImage*, 155:530–548, 2017.

[109] S. Resino, J. A. Seoane, J. M. Bellón, J. Dorado, F. Martin-Sanchez, E. Alvarez, J. Cosín, J. C. López, G. Lopez, P. Miralles, et al. An artificial neural network improves the non-invasive diagnosis of significant fibrosis in hiv/hcv coinfected patients. *Journal of Infection*, 62(1):77–86, 2011.

[110] K. Ristolainen. Predicting banking crises with artificial neural networks: The role of nonlinearity and heterogeneity. *The Scandinavian Journal of Economics*, 120(1):31–62, 2018.

[111] M. Robnik-Šikonja and I. Kononenko. An adaptation of relief for attribute estimation in regression. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML97)*, volume 5, pages 296–304, 1997.

[112] M. Robnik-Šikonja and I. Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine learning*, 53(1-2):23–69, 2003.

[113] B. A. Rocha, A. G. Asimakopoulos, M. Honda, N. L. da Costa, R. M. Barbosa, F. Barbosa Jr, and K. Kannan. Advanced data mining approaches in the assessment of urinary concentrations of bisphenols, chlorophenols, parabens and benzophenones in brazilian children and their association to dna damage. *Environment international*, 116:269–277, 2018.

[114] I. Rodríguez-Ardura and A. Meseguer-Artola. A pls-neural network analysis of motivational orientations leading to facebook engagement and the moderating roles of flow and age. *Frontiers in Psychology*, 11:1869, 2020.

[115] V. P. Romero, L. Maffei, G. Brambilla, and G. Ciaburro. Modelling the soundscape quality of urban waterfronts by artificial neural networks. *Applied Acoustics*, 111:121–128, 2016.

[116] Y. Saeys, T. Abeel, and Y. Van de Peer. Robust feature selection using ensemble feature selection techniques. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 313–325. Springer, 2008.

[117] P. Saikia, R. D. Baruah, S. K. Singh, and P. K. Chaudhuri. Artificial neural networks in the domain of reservoir characterization: A review from shallow to deep models. *Computers & Geosciences*, 135:104357, 2020.

[118] S. Saurabh and K. Dey. Unraveling the relationship between social moods and the stock market: Evidence from the united kingdom. *Journal of Behavioral and Experimental Finance*, page 100300, 2020.

[119] F. Schmitt, R. Banu, I.-T. Yeom, and K.-U. Do. Development of artificial neural networks to predict membrane fouling in an anoxic-aerobic membrane bioreactor treating domestic wastewater. *Biochemical Engineering Journal*, 133:47–58, 2018.

[120] S. S. Selvan, P. S. Pandian, A. Subathira, and S. Saravanan. Comparison of response surface methodology (rsm) and artificial neural network (ann) in optimization of aegle marmelos oil extraction for biodiesel production. *Arabian Journal for Science and Engineering*, 43(11):6119–6131, 2018.

[121] G. C. Sepulveda, S. Ochoa, and J. Thibault. Methodology to solve the multi-objective optimization of acrylic acid production using neural networks as meta-models. *Processes*, 8(9):1184, 2020.

[122] J. Sexton, Y. Everingham, D. Donald, S. Staunton, and R. White. A comparison of non-linear regression methods for improved on-line near infrared spectroscopic analysis of a sugarcane quality measure. *Journal of Near Infrared Spectroscopy*, 26(5):297–310, 2018.

[123] S. Sha, W. Du, A. Parkinson, and N. Glasgow. Relative importance of clinical and sociodemographic factors in association with post-operative in-hospital deaths in colorectal cancer patients in new south wales: An artificial neural network approach. *Journal of evaluation in clinical practice*, 26(5):1389–1398, 2020.

[124] Y. Sharifi, A. Moghbeli, M. Hosseinpour, and H. Sharifi. Study of neural network models for the ultimate capacities of cellular steel beams. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, pages 1–11, 2019.

[125] N. Sharma, K. Dasgupta, and A. Dey. Prediction of natural period of rc frame with shear wall supported on soil-pile foundation system using artificial neural network. *Journal of Earthquake Engineering*, pages 1–25, 2020.

[126] W.-L. Shiau, Y. K. Dwivedi, and H.-H. Lai. Examining the core knowledge on facebook. *International Journal of Information Management*, 43:52–63, 2018.

[127] S. Sinehbaghizadeh, A. Roosta, N. Rezaei, M. M. Ghiasi, J. Javanmardi, and S. Zendehboudi. Evaluation of phase equilibrium conditions of clathrate hydrates using connectionist modeling strategies. *Fuel*, 255:115649, 2019.

[128] V. K. Singh, H. A. Rashwan, S. Romani, F. Akram, N. Pandey, M. M. K. Sarker, A. Saleh, M. Arenas, M. Arquez, D. Puig, et al. Breast tumor segmentation and shape classification in mammograms using generative adversarial and convolutional neural network. *Expert Systems with Applications*, 139:112855, 2020.

[129] F. Soares, M. J. Anzanello, F. S. Fogliatto, M. C. Marcelo, M. F. Ferrão, V. Manfroi, and D. Pozebon. Element selection and concentration analysis for classifying south america wine samples according to the country of origin. *Computers and electronics in agriculture*, 150:33–40, 2018.

[130] D. Strnad, A. Nerat, and Š. Kohek. Neural network models for group behavior prediction: a case of soccer match attendance. *Neural Computing and Applications*, 28(2):287–300, 2017.

[131] F.-S. Tabatabai-Yazdi, A. Ebrahimian Pirbazari, F. Esmaeili Khalilsaraei, N. Asasian Kolur, and N. Gilani. Photocatalytic treatment of tetracycline antibiotic wastewater by silver/tio2 nanosheets/reduced graphene oxide and artificial neural network modeling. *Water Environment Research*, 92(5):662–676, 2020.

[132] P.-N. Tan, M. Steinbach, V. Kumar, et al. *Introduction to data mining*, volume 1. Pearson Addison Wesley Boston, 2006.

[133] S. Tang, J. Lee, S. Loh, and H. Tham. Application of artificial neural network to predict colour change, shrinkage and texture of osmotically dehydrated pumpkin. In *IOP Conf Series: Mat Sci Eng*, volume 206, page 012036, 2017.

[134] Y. Tang, S. Xiao, and Y. Zhan. Predicting settlement along railway due to excavation using empirical method and neural networks. *Soils and foundations*, 59(4):1037–1051, 2019.

[135] F. Tisa, M. Davoody, A. A. A. Raman, and W. M. A. W. Daud. Degradation and mineralization of phenol compounds with goethite catalyst and mineralization prediction using artificial intelligence. *Plos one*, 10(4):e0119933, 2015.

[136] S. Tohidi and Y. Sharifi. Neural networks for inelastic distortional buckling capacity assessment of steel i-beams. *Thin-Walled Structures*, 94:359–371, 2015.

[137] S.-H. Tsaur, Y.-C. Chiu, and C.-H. Huang. Determinants of guest loyalty to international tourist hotelsa neural network approach. *Tourism Management*, 23(4):397–405, 2002.

[138] S. Ullah, B. F. Tanyu, and B. Zainab. Development of an artificial neural network (ann)-based model to predict permanent deformation of base course containing reclaimed asphalt pavement (rap). *Road Materials and Pavement Design*, pages 1–19, 2020.

[139] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore. Relief-based feature selection: Introduction and review. *Journal of biomedical informatics*, 85:189–203, 2018.

[140] R. J. Urbanowicz, R. S. Olson, P. Schmitt, M. Meeker, and J. H. Moore. Benchmarking relief-based feature selection methods for bioinformatics data mining. *Journal of biomedical informatics*, 85:168–188, 2018.

[141] N. N. Vasu and S.-R. Lee. A hybrid feature selection algorithm integrating an extreme learning machine for landslide susceptibility modeling of mt. woomyeon, south korea. *Geomorphology*, 263:50–70, 2016.

[142] M. D. Villas-Boas, F. Olivera, and J. P. S. de Azevedo. Assessment of the water quality monitoring network of the piabanha river experimental watersheds in rio de janeiro, brazil, using autoassociative neural networks. *Environmental monitoring and assessment*, 189(9):439, 2017.

[143] Y. Wang, S. Zhang, Z. Ma, and Q. Yang. Artificial neural network model development for prediction of nonlinear flow in porous media. *Powder Technology*, 373:274–288, 2020.

[144] T. Wong, H. K. Chan, and E. Lacka. An ann-based approach of interpreting user-generated comments from social media. *Applied Soft Computing*, 52:1169–1180, 2017.

[145] T. C. Wong, K. M. Law, H. K. Yau, and S.-C. Ngan. Analyzing supply chain operation models with the pc-algorithm and the neural network. *Expert Systems with Applications*, 38(6):7526–7534, 2011.

[146] T. C. Wong, S. Wong, and K.-S. Chin. A neural network-based approach of quantifying relative importance among various determinants toward organizational innovation. *Expert systems with Applications*, 38(10):13064–13072, 2011.

[147] S. Xie, A. T. Lawniczak, and J. Hao. Modelling autonomous agents decisions in learning to cross a cellular automaton-based highway via artificial neural networks. *Computation*, 8(3):64, 2020.

[148] J. Xu, K. Yamada, K. Seikiya, R. Tanaka, and Y. Yamane. Effect of different features to drill-wear prediction with back propagation neural network. *Precision Engineering*, 38(4):791–798, 2014.

[149] M. Xu, T. C. Wong, and K.-S. Chin. Modeling daily patient arrivals at emergency department and quantifying the relative importance of contributing variables using artificial neural network. *Decision Support Systems*, 54(3):1488–1498, 2013.

[150] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[151] G. H. Yamashita, M. J. Anzanello, F. Soares, M. K. Rocha, F. S. Fogliatto, N. P. Rodrigues, E. Rodrigues, P. G. Celso, V. Manfroi, and P. F. Hertz. Hierarchical classification of sparkling wine samples according to the country of origin based on the most informative chemical elements. *Food Control*, 106:106737, 2019.

[152] D. Yang, J. Kleissl, C. A. Gueymard, H. T. Pedro, and C. F. Coimbra. History and trends in solar irradiance and pv power forecasting: A preliminary assessment and review using text mining. *Solar Energy*, 168:60–101, 2018.

[153] Z. M. Yaseen, S. O. Sulaiman, R. C. Deo, and K.-W. Chau. An enhanced extreme learning machine model for river flow forecasting: State-of-the-art, practical applications in water resource engineering area and future research direction. *Journal of Hydrology*, 569:387–408, 2019.

[154] Y. Yoon, T. Guimaraes, and G. Swales. Integrating artificial neural networks with rule-based expert systems. *Decision Support Systems*, 11(5):497–507, 1994.

[155] O. Yucel, E. S. Aydin, and H. Sadikoglu. Comparison of the different artificial neural networks in prediction of biomass gasification products. *International Journal of Energy Research*, 43(11):5992–6003, 2019.

[156] A. Zeroual, A. Fourar, and M. Djeddou. Predictive modeling of static and seismic stability of small homogeneous earth dams using artificial neural network. *Arabian Journal of Geosciences*, 12(2):16, 2019.

[157] C. Zhang, J. Jiang, J. Ma, X. Zhang, Q. Yang, Q. Ouyang, and X. Lei. Evaluating soil reinforcement by plant roots using artificial neural networks. *Soil Use and Management*, 31(3):408–416, 2015.

[158] H. Zhang, H. Nguyen, X.-N. Bui, T. Nguyen-Thoi, T.-T. Bui, N. Nguyen, D.-A. Vu, V. Mahesh, and H. Moayedi. Developing a novel artificial intelligence model to

estimate the capital cost of mining projects using deep neural network-based ant colony optimization algorithm. *Resources Policy*, 66:101604, 2020.

[159] K. Zhang, B. Zhang, B. Chen, L. Jing, Z. Zhu, and K. Kazemi. Modeling and optimization of newfoundland shrimp waste hydrolysis for microbial growth using response surface methodology and artificial neural networks. *Marine pollution bulletin*, 109(1):245–252, 2016.

[160] L. Zhang, W. Ding, J. Qiu, H. Jin, H. Ma, Z. Li, and D. Cang. Modeling and optimization study on sulfamethoxazole degradation by electrochemically activated persulfate process. *Journal of Cleaner Production*, 197:297–305, 2018.

[161] W. Zhang, L. Jin, E. Song, and X. Xu. Removal of impulse noise in color images based on convolutional neural network. *Applied Soft Computing*, 82:105558, 2019.

[162] X. Zhang, H. Nguyen, X.-N. Bui, H. A. Le, T. Nguyen-Thoi, H. Moayedi, and V. Mahesh. Evaluating and predicting the stability of roadways in tunnelling and underground space using artificial neural network-based particle swarm optimization. *Tunnelling and Underground Space Technology*, 103:103517, 2020.

[163] Z. Zhang, M. W. Beck, D. A. Winkler, B. Huang, W. Sibanda, H. Goyal, et al. Opening the black box of neural networks: methods for interpreting neural network models in clinical applications. *Annals of translational medicine*, 6(11), 2018.

[164] S. A. Ziaee, E. Sadrossadat, A. H. Alavi, and D. M. Shadmehri. Explicit formulation of bearing capacity of shallow foundations on rock masses using artificial neural networks: application and supplementary studies. *Environmental earth sciences*, 73(7):3417–3431, 2015.

[165] C. W. Zobel and D. F. Cook. Evaluation of neural network variable influence measures for process control. *Engineering applications of artificial intelligence*, 24(5):803–812, 2011.

APÊNDICE A

Descrição dos estudos que utilizaram WBFS conforme as cagetorias de uso

Tabela A.1: Estudos que se basearam na categoria B de uso dos WBFS.

Objetivo do estudo	**Qnt. RNA**	**Tipo de RNAs**	**Ref.**
Predição de comportamento em grupo, caso de comparecimento a partidas de futebol	4	MLP. uma rede neural *feedforward* com atraso de tempo, rede neural recorrente, e uma rede neural de base radial. A rede MLP obteve a melhor performance.	[130]
predição da sobrepressão de ar induzida pela explosão em mina a céu aberto	8	Modelos com 1, 2, e 3 camadas ocultas com 5 - 9 neurônios em cada camada. Os resultados de 8 modelos foram reportados, em que a arquitetura 7-9-7-5-1 obteve a melhor performance.	[13]
Predição do risco de fratura de quadril em idosos do sexo feminino e masculino	20	Os autores treinaram 20 modelos para construir um classificador ensemble. O algoritmo de Olden foi aplicado apenas na rede neural mais precisa das 20.	[81]
Avaliação da influência de parâmetros acústicos e visuais objetivos na percepção do ambiente sônico da orla de Nápoles, Itália	500	Os autores reportaram os valores de RMSE e a correlação entre os valores preditos e atuais para os 10 modelos mais acurados. O algoritmo de Olden foi aplicado no melhor modelo.	[115]

Continua na próxima página

Tabela A.1 – continuação da página anterior

Objetivo do estudo	Qnt. RNA	Tipos de RNAs	Ref.
prediction of bearing capacity of square footing resting	7	Sete diferentes números de neurônios ocultos foram testados. O melhor modelo foi obtido com 10 neurônios na camada oculta.	[4]
Predição do custo de capital de projetos de mineração	10	Os autores treinaram 10 arquiteturas de RNA com diferentes números de camadas ocultas e parâmetros do algoritmo de colonização de formigas. O melhor modelo foi obtido com 4 camadas ocultas (5-25-20-18-15-1).	[158]
Predição da estabilidade estática e sísmica de pequenas barragens	17	Os autores treinaram 17 RNAs com diferentes números de NO $(i = 7, 8, \cdots, 23)$. O melhor modelo foi obtido com 21 NO.	[156]
Predição da capacidades de momento final para vigas de aço	12	Os autores treinaram 12 RNAs com diferentes números de NO $(i = 1, 2, \cdots, 12)$. O melhor modelo foi obtido com 1 NO.	[136]
Modelagem de hidrólise de resíduos de camarão para crescimento microbiano.	15	Os autores treinaram 12 RNAs com diferentes números de NO $(i = 2, 3, \cdots, 16)$. O melhor modelo foi obtido com 11 NO.	[159]
Predição do tamanho de microgotículas em sistemas microfluídicos.	10	Os autores treinaram 10 RNAs com diferentes números de NO $(i = 1, 2, \cdots, 10)$. O melhor modelo foi obtido com 10 NO.	[83]
Quantificação da importância relativa da umidade do solo, nitrogênio e temperatura na taxa de hidrólise da ureia	11	Os autores treinaram 22 RNAs com diferentes números de NO $(i = 2, 3, \cdots, 12)$ e dois algoritmos de treinamento (BP, IBP). O melhor modelo foi obtido com 6 NO e IBP.	[78]

Continua na próxima página

Tabela A.1 – continuação da página anterior

Objetivo do estudo	Qnt. RNA	Tipos de RNAs	Ref.
Previsão de mudança de cor, encolhimento e textura de abóbora osmoticamente desidratada	10	Os autores treinaram 10 RNAs com diferentes números de NO ($i = 3,4,\cdots,12$). O melhor modelo foi obtido com 10 NO.	[133]
Predição das propriedades geotécnicas da turfa fibrosa	16	Os autores treinaram 16 RNAs com diferentes números de NO ($i = 1,2,\cdots,16$). O melhor modelo foi obtido com 12 NO.	[28]
Previsão de temperaturas profundas de reservatório usando a composição da fase gasosa de fluidos geotérmicos	455	Os autores treinaram 455 RNAs com diferentes n de NO, variáveis de entrada, e taxa de aprendizado. O melhor modelo foi obtido com a arquitetura 5-9-1, 4-13-1, 3-34-1, 4-10-1, 3-8-1 e 4-8-1. O algoritmo de Garson foi aplicado em cada uma e os resultados foram interpretados de forma individual.	[99]
Predição da frequência de mortes de pedestres na Índia	12	Os autores treinaram 12 RNAs baseadas em 3 funções de ativação e quatro algoritmos de treinamento. O melhor modelo foi obtido com o algoritmo BRNN e a função tansig.	[15]
Predição da mineralização de compostos fenólicos.	24	Os autores treinaram 12 RNAs com base em dois algoritmos de treinamento e seis funções de ativação para modelar dois problemas. Os melhores modelos obtidos foram com LM e as funções tansig e linear.	[135]
Otimização da extração de óleo de sementes de marmelos para produção de biodiesel.	18	Os autores treinaram 18 RNAs com base em três funções de ativação de seis algoritmos de treinamento. O melhor modelo foi obtido com LM e as funções tansig e linear.	[120]

Continua na próxima página

Tabela A.1 – continuação da página anterior

Objetivo do estudo	Qnt. RNA	Tipos de RNAs	Ref.
Avaliação do processo de tratamento de águas residuais	20	Os autores treinaram 20 RNAs com diferentes números de NO ($i = 1,2,\cdots,20$). O melhor modelo foi obtido com 3 NO.	[61]
Predição da deformação permanente da camada de base contendo pavimento asfáltico recuperado	51	Os autores treinaram 51 RNAs com base em diferentes número de NO ($i = 4,5,\cdots,20$) e três funções de ativação. O melhor modelo foi obtido com 9-13-1 e a função tansig.	[138]
Predição da capacidade de adsorção de *Gundelia tournefortii* para a remoção de Pb (II) da solução aquosa	20	Os autores treinaram 20 RNAs com base em diferentes número de NO ($i = 1,2,\cdots,20$). O melhor modelo foi obtido com 12 NO.	[105]
Predição da capacitância de carbono de sacarose dopado com ácido bórico	17	Os autores treinaram 11 RNAs para determinar o melhor algoritmo de treinamento, e em seguida treinaram 6 RNAs para determinar o melhor número de NO. O melhor modelo foi obtido com 21 NO e o algoritmo LM.	[33]
Predição dos padrões de propriedade de veículos domésticos	20	Os autores treinaram 20 RNAs com base em diferentes número de NO ($i = 1,2,\cdots,20$) e 10 valores de *weight decay* para a predição em apenas áreas urbanas, áreas suburbanas e ambas as áreas. O melhor modelo foi obtido com 7, 5, e 19 NO, respectivamente.	[51]
Caracterização do mecanismo de crescimento de uma granulação fundida em leito de fluido in situ	10	Os autores treinaram 10 RNAs com base em diferentes número de NO ($i = 1,2,\cdots,10$). O melhor modelo foi obtido com 5 NO.	[68]
Predição da produção de bio-óleo de pirólise de *Tithonia diversifolia*	16	Os autores treinaram 16 RNAs com base em diferentes número de NO ($i = 1,2,\cdots,16$). O melhor modelo foi obtido com 14 NO.	[11]

Continua na próxima página

Tabela A.1 – continuação da página anterior

Objetivo do estudo	Qnt. RNA	Tipos de RNAs	Ref.
Predição da eficiência da remoção de cloreto de metileno	20	Os autores treinaram 20 RNAs com base em diferentes número de NO ($i = 1,2,\cdots,20$).	[63]
Otimização multi-objetivo da produção de ácido acrílico	10	Os autores treinaram 0 RNAs com base em diferentes número de NO ($i = 1,2,\cdots,10$).	[121]
Predição da capacidade de suporte de sapatas de tiras apoiadas em terreno inclinado	12	Os autores treinaram 12 RNAs com base em diferentes número de NO ($i = 1,2,\cdots,12$). O melhor modelo foi obtido com 9 NO.	[3]
Avaliação das condições de equilíbrio de fase de clatratos hidratados	14	Os autores treinaram 3 RNAs com base em 3 funções de ativação (tansig, sigmoid, elliot-sigmoid). Após determinar a função de ativação, os autores realizaram uma busca pelo número de NO ($i = 5,6,...,15$). O melhor modelo foi obtido com a função tansig.	[127]
Previsão da concentração do traçador e da temperatura de um novo sonorreator de alta frequência	50	Os autores treinaram 25 RNAS para cada problema. Cada grupo de RNAs variou no número de neurônios ocultos ($i = 4,6,8,10,12$) e ($i = 6,8,10,12,14$), e em 5 algoritmos de treinamento. Os melhores modelos foram obtidos com 10 e 12 NO com o algoritmo de LM	[94]
Predição do período natural de estrutura de concreto reforçado	16	Os autores treinaram 16 RNAs com base em diferentes número de NO ($i = 1,2,\cdots,16$). O melhor modelo foi obtido com 6 NO.	[125]
Predição de fluxo não linear em meios porosos como leito compactado de partículas	5	Os autores treinaram 5 RNAs com base em diferentes número de NO ($i = 4,5,\cdots,8$). O melhor modelo foi obtido com 6 NO	[143]

Continua na próxima página

Tabela A.1 – continuação da página anterior

Objetivo do estudo	Qnt. RNA	Tipos de RNAs	Ref.
Predição de desempenho fotocatalítico de tratamento de águas residuais	29	Os autores treinaram 29 RNAs com base em diferentes número de NO ($i = 2,3,\cdots,30$). O melhor modelo foi obtido com 11 NO.	[131]
Predição do assentamento de jangada apoiada na areia reforçada	17	Os autores treinaram 17 RNAs com base em diferentes número de NO ($i = 2,3,\cdots,18$). O melhor modelo foi obtido com 12 NO.	[71]
Predição da capacidade de momento final da viga de aço castelado	8	Os autores treinaram 8 RNAs com base em diferentes número de NO ($i = 1,2,\cdots,8$). O melhor modelo foi obtido com 8 NO.	[56]
Predição da excentricidade do parafuso e da solda de placa única convencional	20	Os autores treinaram 10 RNAs com base em diferentes número de NO ($i = 1,2,\cdots,10$) para cada função função de ativação tansig e sigmoid. O melhor modelo foi obtido com 2 NO e a função tansig.	[54]
Predição da remoção biossortiva de *Nile blue A* de correntes aquosas.	20	Os autores treinaram 8 RNAs com base em diferentes número de NO ($i = 1,2,\cdots,20$). O melhor modelo foi obtido com 12 NO.	[89]
Avaliação do efeito das frequências de carregamento na deformação permanente de materiais granulares não ligados	10	Os autores treinaram 8 RNAs com base em diferentes número de NO ($i = 5,6,\cdots,14$). O melhor modelo foi obtido com 7 NO.	[5]
Predição das capacidades finais das vigas de aço celular	10	Os autores treinaram 10 RNAs com base em diferentes número de NO ($i = 1,2,\cdots,10$). O melhor modelo foi obtido com 7 NO.	[124]

Tabela A.2: Estudos que se basearam na categoria C de uso dos WBFS.

Objetivo do estudo	**Qnt. RNA**	**Tipos de RNAs**	**Ref.**
Predição de subprodutos de desinfecção de água potável	20	Média de 20 RNAs com o algoritmo de Olden.	[96]
Análise da interação das ondas com a estrutura costeira e portuária	500	Média de 500 valores de importância gerados a partir de amostras *bootstrap* para a RNAs com o algoritmo de Garson	[37]
Avaliação dos parâmetros químicos do mel e da temperatura e coeficiente de perda dielétrica do mel	20	Os autores treinaram 300 RNAs com arquiteturas diferentes (NO, funções de ativação e inicialização dos pesos). As 20 redes com melhor capacidade de predição foram escolhidas para aplicar os algoritmos de Garson, PaD e análise de sensitividade.	[97]
Avaliação do reforço do solo por raízes das plantas	7	Os autores utilizaram 7 RNAS com diferentes NO $(i = 2,4,\cdots,14)$ para aplicar o algoritmo de Garson. Os autores reportaram os valores mínimo, máximo, médio e o desvio padrão de importância de cada variável.	[157]
Predição de chuvas durante maio e junho para a bacia do rio Han, Coreia do Sul.	10000	Os autores treinaram uma série de RNAS com 100 amostras de treinamento com 100 conjuntos de pesos iniciais para cada número de NO $(i = 2,3,\cdots,10)$. Uma alta variação na importância média das variáveis foi observada ao utilizar o algoritmo de Olden.	[75]
Avaliação da influência da estrutura e hidráulica do habitat em comunidades tropicais de macro invertebrados	100	Os autores treinaram uma série de RNAs e as 100 melhores foram escolhidas para aplicar o algoritmo de Olden. Na maioria dos casos, o desvio padrão foi maior que a média da importância das variáveis.	[88]

Continua na próxima página

Tabela A.2 – continuação da página anterior

Objetivo do estudo	Qnt. RNA	Tipos de RNAs	Ref.
Avaliação do processo de ultrafiltração aprimorada por micelar para remoção de metais pesados.	1000	Os autores utilizam um grupo de RNAs por meio de uma simulação de Monte Carlo com 1000 execuções para aplicar o algoritmo de Garson.	[79]
Predição de chuva na bacia do rio Geum, Coreia do Sul	100	Os autores utilizaram um conjunto de 100 RNAs para aplicar o algoritmo de Garson e de Olden. Em ambos os casos uma grande variabilidade dos valores de importância foi observada.	[74]
Predição do valor da relação de suporte da Califórnia de algumas areias do mar Egeu	4	Os autores utilizaram 4 RNAs de mesma arquitetura com diferentes inicialização dos pesos para aplicar o algoritmo de Garson.	[32]
Predição de crises bancárias	-	Os autores treinaram diferentes modelos a partir de diferentes NO e valores de weight decay e descrevem os resultados do melhor modelo. Em seguida, os valores de importância de Garson são descritos com a média e o desvio padrão. No entanto, não é claro qual o conjunto de RNAs que foi utilizado para aplicar o algoritmo de Garson.	[110]
Modelagem da degradação do sulfametoxazol por processo de persulfato eletro quimicamente ativado	40	Os autores treinaram 40 RNAs com arquitetura 14-8-1 para aplicar o algoritmo de Garson. A variação obtida nos valores de importância não é descrita, apenas a média.	[160]
Predição do impacto da compactação dinâmica de rolamento usando resultados de teste de penetrômetro de cone dinâmico	4	Os autores treinaram várias RNAs com diferentes NO e escolheram o melhor modelo. Em seguida os autores treinaram 4 RNAs com a melhor arquitetura (8-4-1) obtida para aplicar o algoritmo de Garson.	[107]

Continua na próxima página

Tabela A.2 – continuação da página anterior

Objetivo do estudo	Qnt. RNA	Tipos de RNAs	Ref.
Predição de adsorção de proteína em monocamadas automontadas para a triagem e design de materiais	2000	Os autores utilizam 2000 RNAs com arquitetura 10-6-1 para aplicar o algoritmo de Olden. Em algumas variáveis, o desvio padrão foi maior que a média de importância.	[72]
Análise de fatores clínicos e sociodemográficos em associação com pós-operatório em mortes hospitalares em pacientes com câncer colorretal	50	Os autores utilizaram 50 RNAs com arquitetura 22-14-1 para estimar a importância das variáveis com base no algoritmo de Olden. Os resultados apresentados em forma de boxplot demonstram uma variação nos valores de importância.	[123]
Estimar a radiação solar global de 10 min em plano inclinado arbitrário	8	Os autores utilizaram 8 RNAs com arquitetura 10-10-1 para aplicar o algoritmo de Olden.A variação obtida nos valores de importância não é descrita, apenas a média.	[6]
Análise de orientações motivacionais que levam ao engajamento no Facebook	10	Os autores utilizaram dez RNAs para aplicar o algoritmo de Garson. A variação nos valores de importância não é descrita, apenas a média.	[114]
Predição de precipitação em Jingdezhen, província de Jiangxi, China	30	Os autores treinaram 30 RNAs para cada número de NO $(i = 2,3,\cdots,10)$. Após determinar a melhor arquitetura (9-6-1), o algoritmo de Garson foi aplicado no conjunto de 30 RNAs.	[64]
Predição de assentamento ao longo da ferrovia	4	Os autores treinaram 17 RNAs com diferentes NO para determinar o melhor modelo $(i = 3,4,\cdots,19)$. Em seguida foram utilizadas 4 RNAs de arquitetura 9-12-1 com diferentes inicializações para aplicar o algoritmo de Garson. A variação nos valores de importância não é descrita, apenas a média.	[134]

Continua na próxima página

Tabela A.2 – continuação da página anterior

Objetivo do estudo	Qnt. RNA	Tipos de RNAs	Ref.
Predição da condutividade elétrica do revestimento de filme de nanocompósito de látex de poliestireno dopado com nanotubo de carbono de múltiplas paredes	5	Os autores treinaram 22 RNAs com diferentes NO $(i = 1,2,\cdots,11)$ e duas funções de ativação Após determinar o melhor modelo, os autores treinaram 5 RNAs de arquitetura 5-8-1 com a função sigmoid para aplicar o algoritmo de Garson. A variação nos valores de importância não é descrita, apenas a média.	[29]
Predição da resposta fisiológica de corações de *Tivela stultorum* com digoxina	20	Os autores treinaram 20 modelos para 12 diferentes arquitetura de RNA (diferentes NO e algoritmos de treinamento). A média dos valores de importância determinado a partir das 20 RNAs foi apresentada para cada arquitetura.	[36]
Predição de propriedades dielétricas do mel	20	Diferentes arquiteturas de RNAs foram treinadas e as 20 melhores foram utilizadas como base para o algoritmo de Garson e o método PaD. A variação nos valores de importância não é descrita, apenas a média.	[98]