

## AVALIAÇÃO DA QUALIDADE DE FERRAMENTAS PARA O ENSINO DE REDES NEURAIS EM AMBIENTE EDUCACIONAL

Valderêdo Sedano Fontana - Multivix<sup>1</sup>

Ednea Zandonadi Brambila Carletti - Multivix<sup>2</sup>

### RESUMO

O artigo apresenta uma análise das ferramentas computacionais NeuroFurg, GMDH Shell DS e Weka, ferramentas voltadas para a formação de estudantes da área de Ciência da Computação, com foco no estudo da Inteligência Artificial, especificamente, Redes Neurais Artificiais (RNAs), usando modelos de neurônios Perceptron, K-fold e SimpleKmeans. Os softwares apresentados possibilitam a construção de modelos destes três tipos de neurônios em diversos casos. Tendo em vista o uso aplicado ao ensino, após a apresentação das ferramentas, o artigo procurou avaliar características pedagógicas e ergonômicas por meio de uma lista de verificação integrada (checklist-ergolist) pré estabelecida. Os resultados levam a professores e alunos escolherem softwares para atividades técnicas específicas e conseguir um melhor resultado na aprendizagem.

**Palavras-chave:** Redes Neurais. Ergonomia em Software. Avaliação Checklist. Softwares de Redes Neurais.

### ABSTRACT

The article presents an analysis of computational tools NeuroFurg, GMDH Shell DS and Weka tools aimed at training students in the field of Computer Science, focusing on the study of Artificial Intelligence, specifically, Artificial Neural Networks (ANN), using neuron models Perceptron, K-fold and SimpleKMeans. The software presented, allow the construction of models of these three types of neurons in several cases. Given the applied use at the school, after the presentation of the tools,

---

1 Mestrando em Pesquisa Operacional e Inteligência Computacional (UCAM). Especialista em Informática na Educação (IFES). Especialista em Gestão Empresarial (FACEL). Bacharel em Ciência da Computação, Licenciado em Física e Pedagogia. Professor da Faculdade Multivix Cachoeiro de Itapemirim.

2 Doutoranda em Ciências da Educação pela Universidade Autônoma de Assuncion (UAA). Mestre em Ciência da Informação (PUC-CAMPINAS). Especialista em Informática na Educação (IFES). Graduada em Pedagogia (FAFIA). Professora e Coordenadora de Pesquisa e Extensão da Multivix Cachoeiro de Itapemirim.

the article sought to evaluate pedagogical and ergonomic characteristics through an integrated checklist (checklist-ergolist) pre established. The results lead to teachers and students choose software for specific technical activities and achieve a better result in learning.

**Keywords:** Neural networks. Software ergonomics. Checklist evaluation. Neural Networks Software.

## 1 INTRODUÇÃO

Com a maior utilização de recursos de informática na sociedade e em ambientes educacionais, percebe-se o aumento da utilização de *softwares* com fins educativos. A evolução das tecnologias está motivando a educação para novos rumos, pois dessa forma é possível propiciar evolução no processo de ensino/aprendizagem.

A utilização de Tecnologias de Informação e Comunicação (TIC) mostra-se como uma opção favorável a um processo de ensino-aprendizagem mais apropriado à formação do aluno. No entanto, o emprego destas tecnologias para fins educacionais necessita de análise e estudo.

A motivação para realizar deste estudo surgiu da necessidade que professores e alunos de Computação têm de facilitar a aprendizagem de Redes Neurais, tornando o aluno sujeito mais motivado e como consequência, fazer com que os discentes desenvolvam competências que resultem na aplicação prática da ferramenta visando a melhor qualificação.

A Engenharia de Usabilidade, termo criado por Jakob Nielsen em seu livro publicado de mesmo nome em 1996, representa todo um esforço lógico para garantir a qualidade em uso de programas de software interativo como foco na educação.

Segundo Oliveira (2004), na visão da qualidade do software, a usabilidade é um conjunto de atributos que evidenciam o esforço necessário para sua utilização, bem como o julgamento individual deste uso, por um grupo de usuários.

Dessa forma, torna-se cada vez mais necessária a verificação da real contribuição

técnica e pedagógica do impacto do uso dessas ferramentas computacionais criadas para o ambiente de aprendizagem, de modo que seja possível, através de testes específicos, escolher uma ferramenta em relação à outra, melhorando o processo de ensino.

Escolher um software e classificá-lo quanto à eficiência e eficácia que o mesmo dispõe em ambiente pedagógico, em qualquer nível de ensino, é uma tarefa complexa, pois envolve diversas análises que vão desde o processo de entrada dos dados, seguindo do processamento e analisando a saída desses dados, até mesmo, em alguns casos, a etapa de retroalimentação, processo necessário a certos softwares que trabalham com processamento de dados, caso específico desse artigo. Os critérios mais observados nesse tipo de ferramenta devem perpassar a qualidade ergonômica e pedagógica, mostrando-se necessário ao usar a ferramenta na formação de alunos, avaliar a conformidade de critérios de utilizabilidade (facilidade de utilização) e, principalmente, os critérios de natureza didática e psicopedagógico contidos nos programas. A seleção e a avaliação de software é uma tarefa educativa e ao mesmo tempo técnica que exige a compreensão do contexto educacional.

A aplicação prática do *checklist-ergolist* avaliou o desempenho didático e ergonômico dos programas *NeuroFurg*, *GMDH Shell DS* e *Weka*, que tem aplicação para o ensino de computação. Como resultados são apresentados os valores de qualidade conseguidos pelo programa dentro dos critérios utilizados na avaliação e as conclusões a respeito do método (*checklist-ergolist*) e dos resultados obtidos.

Importante observar que a avaliação da qualidade de software tanto na construção de projetos, quanto de produtos, está regulamentada pelas normas: ISO 9126 (qualidade de produto), ISO 12119 (qualidade de pacotes - software de prateleira, vendido como um produto embalado), ISO 14598 (guia para avaliação de produtos) e ISO 12207 (qualidade do processo de desenvolvimento).

Este trabalho tem como objetivo, demonstrar por meio de testes de simulação de qualidade, três ferramentas existentes para o ensino de Redes Neurais Artificiais, área da Ciência da Computação, o alinhamento de questões pedagógicas e

ergonômicas com os conceitos já desenvolvidos e experimentados da ergonomia de software. Foi aplicada uma lista de verificação- *checklist-ergolist* aos softwares em selecionados e estudados.

A lista aplicada foi desenvolvida junto ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina. Dentre as técnicas que podem orientar e demonstrar parâmetros para o processo de avaliação de ferramentas informatizadas de educação pode-se destacar o uso de *checklist-ergolist*. Embora limitado, é uma das formas de avaliação rápida e de baixo custo. No caso de avaliações de software educacional, o *checklist-ergolist* fornece uma lista de perguntas voltadas para o interesse pedagógico e à usabilidade (facilidade de uso) dos programas, as quais ajudam a focalizar os critérios a serem avaliados (SQUIRE; PREECE, 1996).

## 2 ABORDAGEM PEDAGÓGICA

Nielsen (2005) considera que usabilidade é quando um *software* seja de fácil manuseio e de rápida aprendizagem. Conclui-se ainda que este produto não deva ser facilmente esquecido, não provoque erros operacionais e ofereça alto grau de satisfação para os usuários, resolvendo de forma dinâmica os propósitos para o qual foi projetado.

Outra definição para a palavra em contexto de software é dita pela norma ISO 9241-11 (1998), onde descreve que a usabilidade é a capacidade de uso de um produto por um grupo específico de usuários, em um contexto específico.

Ao usarmos ferramentas com propósitos pedagógicos, a avaliação da qualidade de um *software* educacional é extremamente importante para que os objetivos de aula sejam alcançados. Assim, devem-se levar em conta, principalmente, as características relacionadas à qualidade didático-educativa. Nessa integração da computação com os estudiosos de ergonomia, é possível cabe aos educadores questionar se os programas em ambiente de sala de aula, realmente agregam valor ao aprendizado do aluno. Os professores precisam conhecer as bases dos preceitos pedagógicos, das ideologias e das teorias psicológicas desenvolvidas embutidas na

elaboração dos programas. Os responsáveis pela educação devem questionar a aplicabilidade desses programas como ferramentas didático-pedagógicas e o seu nível de qualidade.

Para Silva (1998) uma solução para a verificação avaliada de *softwares* de educacionais seria aliar os conceitos da ergonomia de *software* com a prática educacional, sendo importante aplicar esses conhecimentos tanto para avaliar a qualidade de um produto já existente (lançado no mercado) como ao longo do seu desenvolvimento (processo de produção).

Dentre as ferramentas que podem orientar e fornecer parâmetros para o processo de avaliação está à utilização de *checklist-ergolist*. Embora limitado, é uma das formas de avaliação rápida e de baixo custo. No caso de avaliações de *software* educacional, o *checklist-ergolist* fornece uma lista de perguntas voltadas para o interesse pedagógico e à usabilidade (facilidade de uso) dos programas, as quais ajudam a focalizar os critérios a serem avaliados (SQUIRE; PREECE, 1996).

Quanto ao uso de qualquer ferramenta que se pretenda utilizar em ambiente de ensino, a qualidade é determinante, sendo a seleção e a avaliação de *software* é uma tarefa educativa e ao mesmo tempo técnica que exige o entendimento do contexto educacional. Os critérios pedagógicos que devem ser avaliados *nos mesmos* são: os objetivos, os conteúdos, a didática, a capacidade interativa e a apresentação dos conteúdos.

### 3 REDES NEURAIS

Em 1943 foi publicado o primeiro estudo sobre Redes Neurais, *A Logical Calculus of the Ideas immanent in Nervous Activity*, desenvolvido por McCulloch e Pitts, que propunha um modelo de neurônio artificial que poderia ser usado como uma unidade de processamento simples (modelo MCP). Este estudo é até hoje considerado como uma referência importante para o estudo básico de Redes Neurais Artificiais (BEALE; JACKSON, 1990). A rede modelada possuía neurônios simples, que dispunham apenas de duas entradas, excitatórias e inibitórias. O neurônio era ativado no momento que o número de entradas excitatórias ultrapassasse o de

entradas inibitórias, produzindo assim uma saída igual a 1. Para determinar se o neurônio responde ou não a uma entrada, é utilizado o valor da limiar (threshold), que não conta no estudo. Desta forma, quando a limiar fosse menor que o total da entrada, o neurônio seria excitado e propagaria o estímulo que foi recebido. Por outro lado, se a limiar fosse maior que o total da entrada, este permaneceria de forma inibida (LUGER; STUBBLEFIELD, 1989).

Segundo Rezende (2003) Pitts e McCulloch propunham no seu modelo que o armazenamento e o processamento das informações seriam realizados de forma distribuída e paralela, por neurônios artificiais que correspondem a elementos processadores simples.

### **3.1 Redes Neurais Artificiais**

As Redes Neurais Artificiais (RNAs) podem ser definidas como redes de processamento que contem unidades interconectadas chamadas de neurônios artificiais, onde cada uma destas atua de forma específica, podendo ser de entrada/saída e adquire conhecimento através da experiência (CASTRO; VON ZUBEN, 2003).

Segundo Rezende (2003), as RNAs podem ser consideradas como as principais técnicas de simulação do comportamento inteligente, baseadas no funcionamento do cérebro humano, apresentando principalmente as seguintes características:

- Possibilidade de aprendizado por meio de treinamento, utilizando de uma base modelo para ganhar experiência e reconhecer outras instâncias similares que ainda não haviam sido apresentadas.
- Demonstra um bom desempenho quando utilizada em tarefas mal definidas, quando não se sabe como encontrar uma solução, por faltar de conhecimento.
- Resistente contra ruídos, ou seja, não perde desempenho com a falta de informações ou com informações erradas.

As RNAs podem ser aplicadas principalmente nos problemas de aproximação, predição, classificação, categorização e otimização (RUSSELL; NORVIG, 2004). Neste artigo foram utilizados os algoritmos de classificação perceptron e K-fold além do simpleskmeans.

### 3.2 Perceptron

O modelo perceptron, proposto em 1958 por Rosenblatt, incluiu no neurônio a limiar e propôs também um processo que permitiria a rede passar a aprender, fazendo com que os pesos de suas sinapses fossem regulados de forma sistemática. Rosenblatt desenvolveu o perceptron com a estrutura mais simples de redes neurais, sendo assim, capaz de classificar padrões linearmente separáveis. Portanto, durante o processo de treinamento do perceptron, espera-se encontrar um conjunto de pesos que defina uma reta que separe as diferentes classes, de forma que a Rede classifique corretamente as entradas apresentadas.

O modelo possui algumas similaridades com o modelo MCP e, além delas, incluiu ao seu neurônio a limiar e propôs também um processo que permitiria a rede passar a aprender, fazendo com que os pesos de suas sinapses fossem regulados de forma sistemática. Outra diferença é a sua capacidade de receber números reais como entrada, onde no MCP só era possível receber entradas binárias.

A principal característica do perceptron é ser um separador linear, conseguindo assim distinguir e separar padrões entre lados opostos de uma reta. É considerada a forma mais simples de RNA e possui um único neurônio para fazer a classificação de padrões, que podem ter no máximo duas hipóteses (ROSENBLATT, 1958).

### 3.3 K-fold

O método de validação K-fold (*cross-validation* ou *percentage split*) é muito utilizado para a resolução de problemas de classificação, nele, a base de dados é dividida em k subconjuntos. Um destes k subconjuntos é separado para fazer a validação do modelo. Os outros k-1 subconjuntos são utilizados para teste, ou seja, servirão de treinamento para o modelo. Então o sistema aprende com estes subconjuntos de

treinamento e extrai o conhecimento destes. Logo após, o subconjunto retirado para a validação é empregado para analisar e medir a precisão do conhecimento adquirido. A figura 1 demonstra a divisão em subconjuntos.

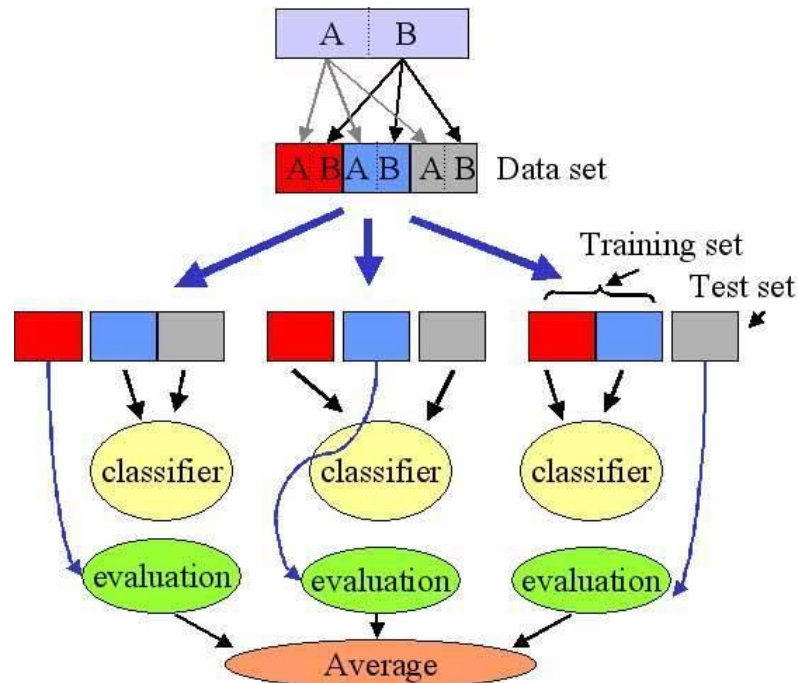


Figura 1 - Exemplo de divisão dos subconjuntos do método k-fold, com k=3.

Fonte: Babilomics Tutorial, acesso em

Na figura 1, é possível visualizar um exemplo de como é feita a separação dos subconjuntos de uma base de dados, tendo  $K=3$ . A validação é criada a partir pedaços dos subconjuntos, sendo cada pedaço pego em sequencia, ou seja, no primeiro subconjunto pegaria a primeira parte, no segundo a segunda e no terceiro a terceira. Já o subconjunto de teste (estimação), é montado com as partes restantes dos subconjuntos. A partir daí a estimação é aprendida e validada pela parte da validação e o resultado (acurácia) é encontrado.

### 3.4 Simplekmeans

O SimpleKMean é um algoritmo que cria e organiza *clusters* fazendo uso da média aritmética, cria-se assim a quantidade de *clusters* solicitada pelo usuário no momento da modelagem. Ao fazer uso de média aritmética, podem-se encontrar valores que não foram testados. Dessa forma, a vantagem de encontrar dados que não foram testados pelo pesquisador é que esses trazem novas possibilidades e visões ao pesquisador, que agora tem uma sugestão para um futuro experimento e,



assim, tentar melhorar a produtividade do problema que deseja resolver. No entanto, em alguns casos é necessário fazer um novo experimento para confirmar estes dados, e ter certeza da real resposta.

#### 4 CONSIDERAÇÕES SOBRE OS MODELOS DESENVOLVIDOS NOS SOFTWARES AVALIADOS

Para a aplicação dos métodos e algoritmos foram utilizadas duas bases de dados e três softwares. A primeira base foi utilizada no software NeuroFURG, esta foi apresentada no artigo NeuroFURG: uma ferramenta de apoio ao ensino de Redes Neurais Artificiais em 2011 por Madsen e Adamatti. A segunda é conhecida como Iris flower dataset ou Fisher's Iris dataset, sendo apresentada por Fisher em 1936.

A seguir uma breve descrição sobre as ferramentas utilizadas de modo que se possa visualizar na prática o real funcionamento.

##### 4.1 Apresentação do Software NeuroFURG

O NeuroFURG, é uma ferramenta desenvolvida para auxiliar no aprendizado de RNAs. Ele dispõe de duas estruturas de neurônios, o modelo o Adaline e o Perceptron, que será tratado neste artigo. A figura 2 apresenta o sistema NeuroFURG com uma base já aplicada.

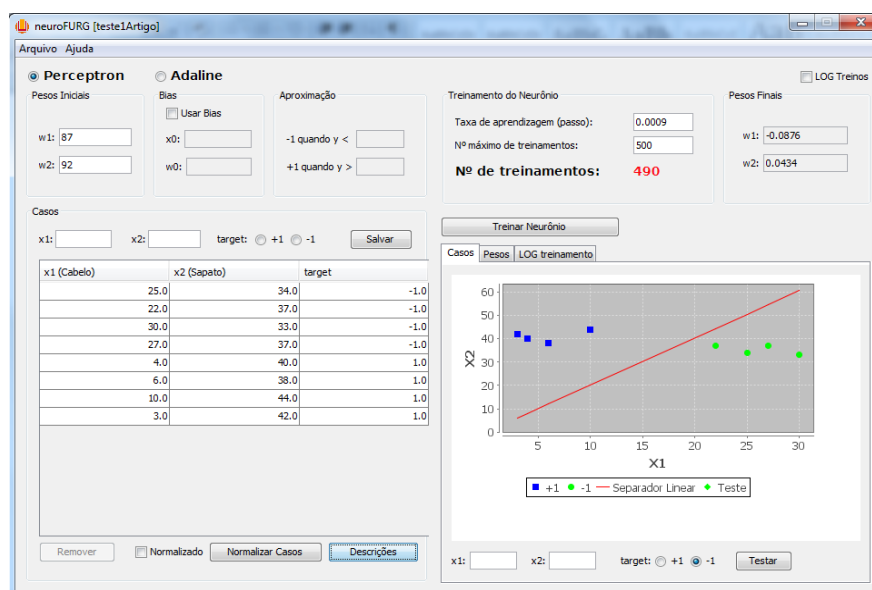


Figura 2 - Ferramenta NeuroFURG.

Fonte: Pesquisa do autor

Na figura 2 foram adicionados os dados cabelos, como  $x_1$ , sapato, como  $x_2$  e sexto, sendo -1 para feminino e +1 para masculino. Para os pesos iniciais, foram utilizados valores aleatórios ( $w_1$  e  $w_2$ ) para serem testados e alterados até chegar aos pesos ideais (descrito pelos valores de  $w_1$  e  $w_2$  dos pesos finais) que são gerados pela ferramenta. Já a taxa de aprendizagem é a porcentagem de aprendizado que a ferramenta terá cada vez que o código for rodado, onde neste caso foi dado um máximo de treinamentos de 500 (número de vezes que o código rodará). É possível visualizar também que foram necessários 490 treinamentos para a ferramenta aprender completamente a base. O software gera também um gráfico, demonstrando os dados separados em grupos pelo separador linear.

Uma característica importante do sistema NeuroFURG é o fato deste só aceitar neurônios com duas entradas, com o objetivo de facilitar a visualização do separador linear relacionado a função de ativação resultante do treinamento (MADSEN; ADAMATTI, 2011).

Outra funcionalidade muito importante do NeuroFURG é a possibilidade de geração do código do neurônio e do treinamento nas linguagens C, JAVA e PHP, o que permite que o código seja explorado e alterado de acordo com a necessidade.

A ferramenta se destaca pela facilidade de utilização, podendo ser facilmente configurado, que se dá pelos valores dos pesos, taxa de aprendizado e número máximo de treinamento. E tem como uma desvantagem a impossibilidade de importação de dados, ou seja, todos os dados devem ser inseridos um a um no programa.

## **4. 2 Apresentação do Software NeuroFURG**

O software GMDH Shell DS é uma ferramenta paga de RNA que capaz de resolver problemas de previsão, classificação e de mineração de dados (Data Mining). Foi desenvolvido com a finalidade de facilitar na criação das RNAs e na obtenção dos resultados, evitando complexidades técnicas.

Uma grande vantagem do software é que os dados não precisam ser previamente normalizados para que sejam utilizados, sendo esta feita pelo software. Ele também aplica o treinamento e a validação automaticamente, apresentando gráficos, e dados como precisão, quantidade de instâncias corretas e incorretas, de forma intuitiva e direta, agilizando assim a interpretação do resultado.

O trabalho irá abordar o modelo de classificação no software GMDH Shell DS, utilizando a base de dados Iris flower, que irá aplicar a metodologia K-fold, onde parte da base será usada para treinamento e a outra para validação.

Para que a classificação da base seja feita, o software apresenta 4 passos simples e fáceis, onde é possível preparar a classificação. As figuras 3, 4, 5 e 6 representam tais passos.

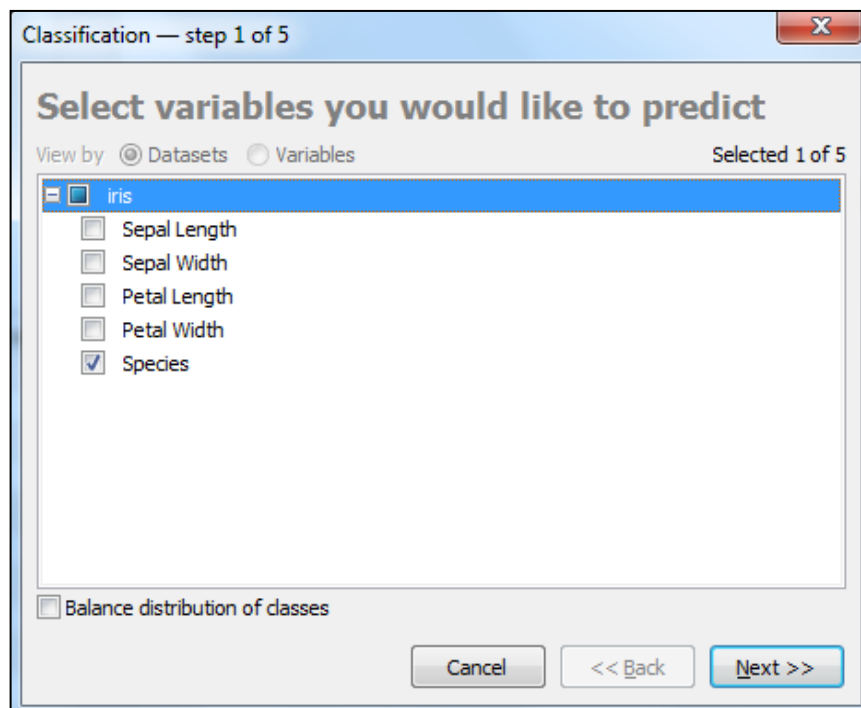


Figura 3 - Passo 1, escolha da variável que será prevista no software GMDH Shell DS.

Fonte: Pesquisa do autor

Na figura 3, é possível escolher o que se quer prever ou classificar, ou seja, com o treinamento e a validação, será possível encontrar os valores da variável escolhida, que no caso foi a espécie (species).

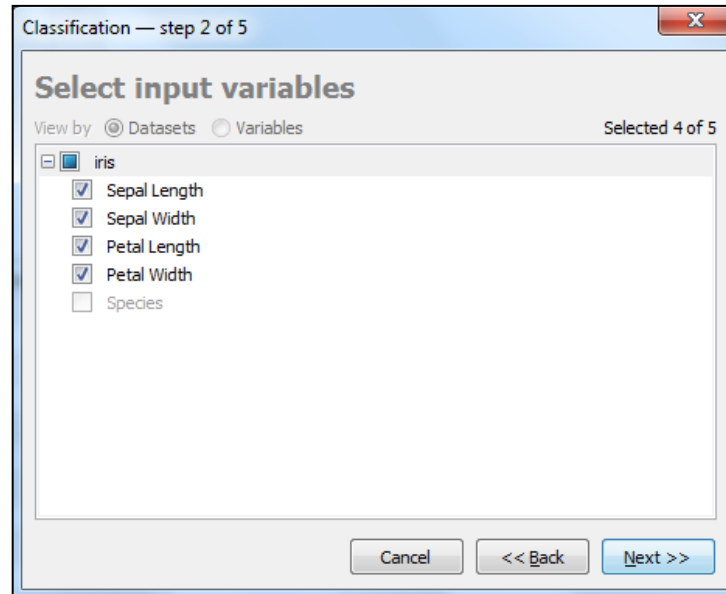


Figura 4: Passo 2, escolha dos dados de entrada no software GMDH Shell DS.

Fonte: Pesquisa do autor

A figura 4 apresenta a escolha dos dados que podem ser utilizados como entrada (input). No exemplo, foram escolhidos todos os demais campos, que serão utilizados para o treinamento e validação para definir quais serão os dados previstos, no caso, o software irá aprender com os dados do treinamento e irá testá-los com os dados da validação (que não existem no treinamento) e prever qual a espécie da flor.

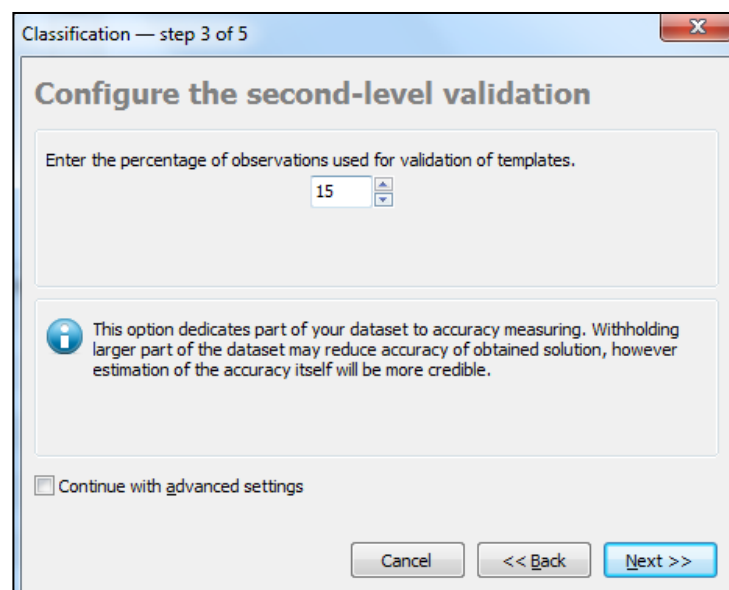


Figura 5 - Passo 3, escolha da porcentagem em que a base de dados será dividida entre o treinamento e a validação no software GMDH Shell DS.

Fonte: Pesquisa do autor

Na figura 5 é escolhida qual a porcentagem da base será separada para a validação. Para o exemplo em questão, foi escolhido fazer uma separação de 15% para a validação.

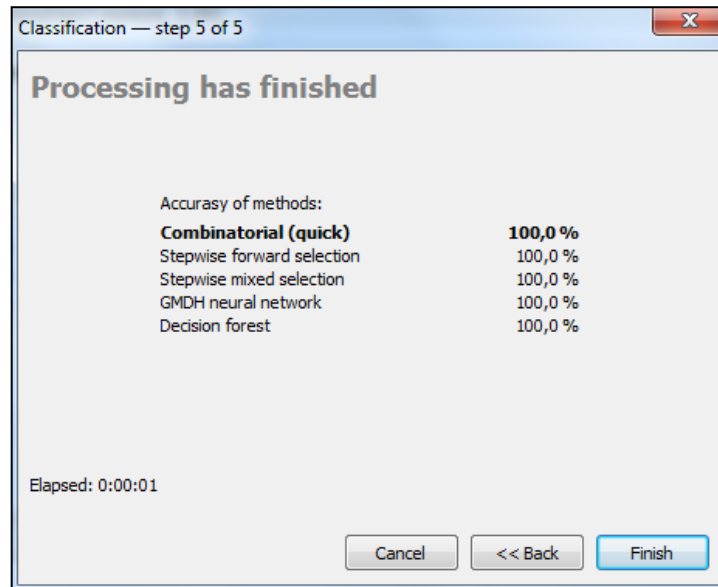


Figura 6 - Passo 4, processamento dos dados no software GMDH Shell DS

Fonte: Pesquisa do autor

A figura 6, o software apresenta a porcentagem de acertos que foi encontrada através da análise e treino. Após as escolhas de da variável que será prevista, dos dados de entrada, da porcentagem que será utilizada para a validação e o processamento do software, ele apresenta os dados do treinamento, da validação e ainda um gráfico da performance da classificação, que são vistos nas figuras 7, 8 e 9.

		Predicted class			Total	Recall
		setosa	versicolor	virginica		
Actual class	<b>setosa</b>	43	0	0	43	1,000
	<b>versicolor</b>	0	41	1	42	0,976
	<b>virginica</b>	0	0	42	42	1,000
	<b>Total</b>	43	41	43	127	
	<b>Precision</b>	1,000	1,000	0,977		
	<b>F-measure</b>	1,000	0,988	0,988		
	<b>Baseline</b>	0,661	0,669	0,669	0,339	
	<b>Accuracy</b>	1,000	0,992	0,992	0,992	

Figura 7 - dados do treinamento feito no software GMDH Shell DS.

Fonte: Pesquisa do autor

A figura 7 apresenta os dados referentes ao treinamento feito pelo software com parte da base. Para o treinamento, foram utilizadas 127 instâncias das 150 totais que continham na base, que é aproximadamente 85% da base. Destas 127, 126 foram classificadas corretamente e apenas uma incorretamente, obtendo um total de 99,2% de acertos no treinamento. Logo abaixo é possível ver uma tabela com as 3 espécies de flor e o número de instancias classificadas em cada uma delas. É visível também na tabela onde ocorreu o erro, uma flor que na verdade era da espécie versicolor foi classificada erroneamente como virginica.

		Predicted class			Total	Recall
		setosa	versicolor	virginica		
Actual class	setosa	7	0	0	7	1,000
	versicolor	0	8	0	8	1,000
	virginica	0	0	8	8	1,000
	Total	7	8	8	23	
<b>Precision</b>		1,000	1,000	1,000		
<b>F-measure</b>		1,000	1,000	1,000		
<b>Baseline</b>		0,696	0,652	0,652	0,348	
<b>Accuracy</b>		1,000	1,000	1,000	1,000	

Figura 8 - dados da validação feita no software GMDH Shell DS.

Fonte: Pesquisa do autor

Na figura 8, é possível observar os dados que foram classificados e previstos na validação feita pelo software. A validação contou com um total de 23 instâncias, que é aproximadamente 15% da base, como escolhido anteriormente. Todas as 23 instâncias foram classificadas corretamente. Resultado que também pode ser visto claramente na tabela gerada pela ferramenta.

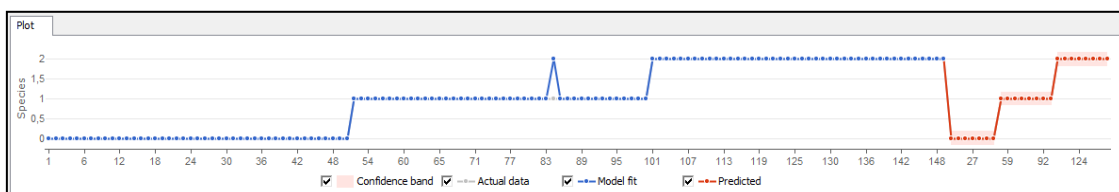


Figura 9 - Gráfico gerado no software GMDH Shell DS, mostrando em azul o aprendizado durante o treinamento e em vermelho a classificação feita na validação.

Fonte: Pesquisa do autor

A figura 9 apresenta o gráfico gerado, que mostra a evolução tanto do treinamento (em azul) quanto da validação (em vermelho), que chegou numa classificação 100% correta.

### 4.3 Apresentação do Software NeuroFURG

A ferramenta WEKA é formada por um conjunto de bibliotecas desenvolvidas em Java de Knowledge-discovery in databases (KDD). Foi desenvolvido na Universidade de Waikato na Nova Zelândia e possui uma grande quantidade de algoritmos para a preparação dos dados, aprendizagem de máquina, mineração de dados e de validação de resultados.

Segundo Machado et al (2011), o processo KDD tem como objetivo extrair informações de bases de dados, para que sejam gerados padrões que antes eram desconhecidos.

Para o teste da base de dados Iris flower, foi utilizado o algoritmo SimpleKMeans no software, como mostra a figura 10.

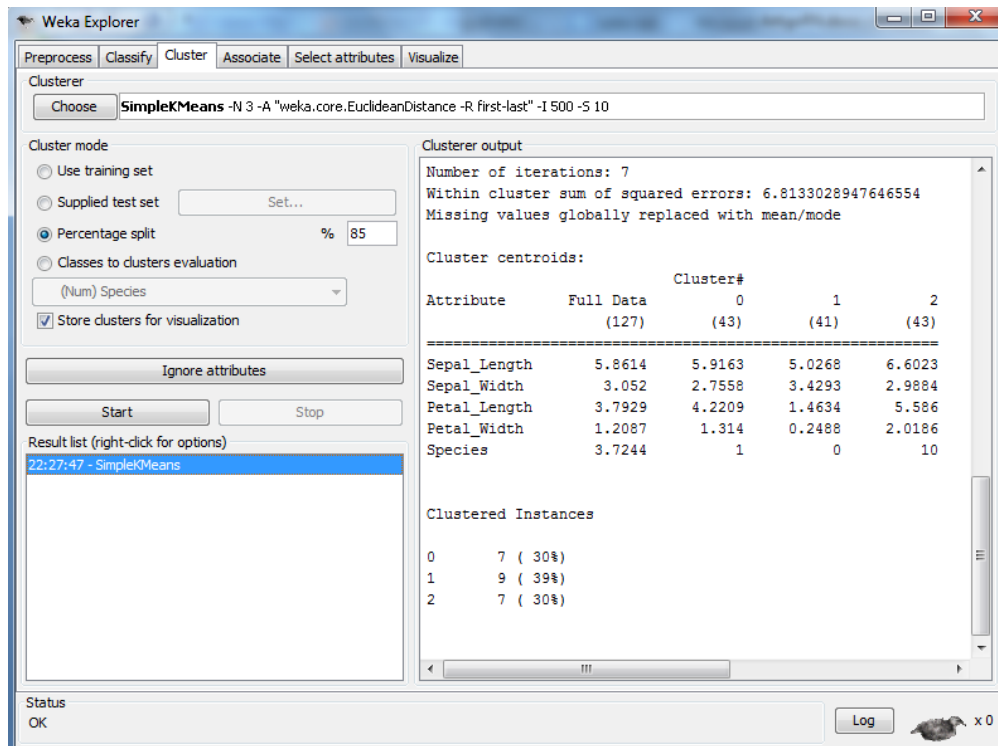


Figura 10 - Utilização do algoritmo SimpleKMeans na ferramenta WEKA.

Fonte: Pesquisa do autor

Na figura 10 é possível observar a utilização do algoritmo SimpleKMeans. Foi escolhido utilizar 85% da base de dados para o treinamento, deixando 15% desta para a validação. Ao final do relatório gerado pela ferramenta WEKA podemos encontrar as instâncias clusterizadas, ou seja, quais foram às classificações geradas, considerando "0" como setosa, "1" como versicolor e "2" como virginica. Além disso, se somarmos as porcentagens da classificação gerada dará um total de 99%, logo, houve 1% de erro na classificação.

## 5 MÉTODO DE VERIFICAÇÃO CHECKLIST-ERGOLIST E RESULTADOS ENCONTRADOS

Os modelos e abordagens para concepção e avaliação de *softwares educacionais* são uma amostra da amplitude e complexidade da tarefa para se estabelecer parâmetros gerais para conceber e avaliar a qualidade pedagógica e ergonômica de programas. O *checklist-ergolist* é composto por perguntas que direcionam o reconhecimento de objetivos que qualificam o *programa* a partir das características desejáveis.

Dessa forma, o percentual que cada tópico apresenta significa o grau de satisfação em relação a essas características. As questões que o teste de verificação *checklist-ergolist* não se aplicou se deu devido à tipologia do programa e não foram consideradas na avaliação.

O teste e questionário (Ergolist) realizado é disponibilizado pelo LabUtil - Laboratório de Utilizabilidade da Informática da Universidade Federal de Santa Catarina, um sistema de listas de verificação de qualidades ergonômicas do software desenvolvido em 1997, com o apoio da Fundação Softex.

Entende-se como "Critérios Ergonômicos" o conjunto de qualidades ergonômicas que as interfaces humano-computador devem apresentar. Eles foram desenvolvidos pelos pesquisadores, Dominique Scapin e Christian Bastien, ligados ao INRIA (Institut National de Recherche en Informatique et en Automatique da França) em 1993. O conjunto de perguntas é composto opções presente em dezoito grupos de



questões que avaliam tópicos de critério especializado da ergonomia de uma interface-humano computador.

Os critérios que identificam os grupos de perguntas usada neste artigo são: presteza, agrupamento por localização, agrupamento por formato, feedback, legibilidade, concisão, ações mínimas, densidade informacional, ações explícitas, controle de usuário, flexibilidade, experiência do usuário, proteção contra erros, mensagens de erro, correção de erros, consistência, significados e compatibilidade.

Na tabela 1 abaixo é apresentado os resultados dos testes nos três softwares de Redes Neurais avaliados quanto aos critérios ergonômicos descritos.

TABELA 1 - LAUDO CHECKLIST - ERGOLIST

Softwares Avaliados de Redes Neurais	Questões Checklist	Critérios Ergonômicos																		
		Concisão	Mensagens de erro	Flexibilidade	Legibilidade	Significados	Proteção contra erros	Agrupamento por formato	Experiência do Usuário	Presteza	Controle do Usuário	Correção de Erros	Consistência	Agrupamento por Localização	Densidade informacional	Feedback	Compatibilidade	Ações explícitas	Ações Mínimas	TOTAL
NEUROFURG	Questões Conformes	12	8	0	19	10	1	12	5	10	4	0	11	11	9	8	8	2	3	133
GMDH		13	8	3	21	11	1	14	5	13	4	2	11	16	9	8	12	3	3	157
WEKA		9	6	0	19	9	1	12	3	8	4	0	10	10	9	8	8	2	3	121
NEUROFURG	Questões Não Conforme	1	1	3	2	2	4	4	1	5	0	5	2	11	9	8	8	2	3	71
GMDH		0	1	0	2	1	4	2	1	2	0	3	2	6	9	8	4	1	3	49
WEKA		4	3	3	6	3	4	4	3	7	0	5	3	12	9	8	8	2	3	87
NEUROFURG	Questões Não Aplicáveis	1	0	0	5	0	2	1	0	2	0	0	0	0	0	1	4	1	0	17
GMDH		1	0	0	5	0	2	1	0	2	0	0	0	0	0	1	4	1	0	17
WEKA		1	0	0	5	0	2	1	0	2	0	0	0	0	0	1	4	1	0	17
Nº de questões		14	9	3	26	12	7	17	6	17	4	5	13	22	18	17	20	5	6	

Fonte: Pesquisa do autor

Através dos valores dos resultados da tabela 1, puderam-se analisar pontos que precisam ser melhorados para tornar os três *softwares* mais usual, ou seja, aprimorar o grau de sua usabilidade, aumentando assim, a satisfação do aluno e a possibilidade de uma aprendizagem mais motivadora e significativa.

Dessa forma, seguindo os preceitos do teste do grau de usabilidade aplicado ao ambiente educacional, é possível, por meio de uma visualização gráfica, conforme o gráfico da figura 11 a seguir, escolher uma ferramenta para adotar em ambiente

educacional.

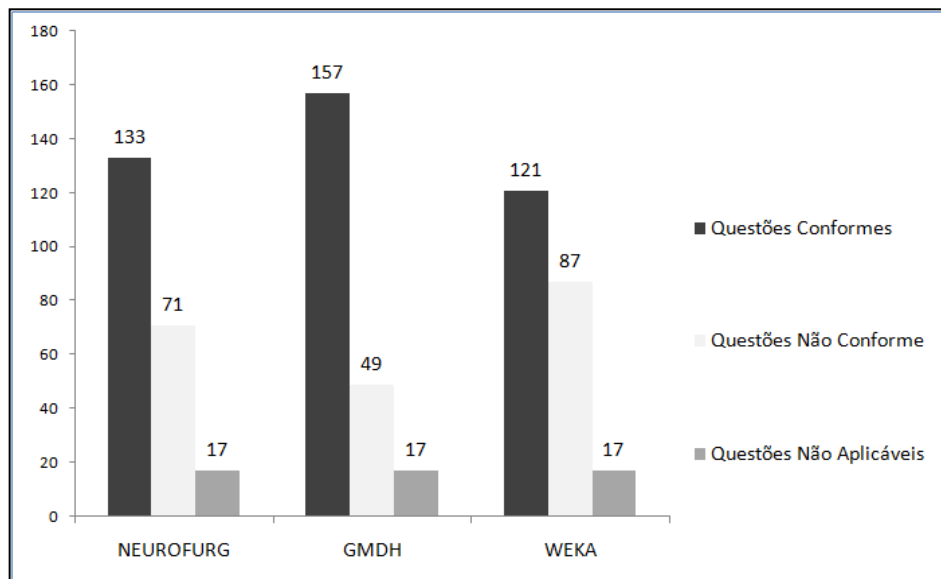


Figura 11 - Gráfico comparativo dos laudos das ferramentas por meio do teste Checklist.

Fonte: Pesquisa do autor

Embasando-se nos resultados apresentados no gráfico da figura 11 acima, observa-se que a usabilidade do *software* GMDH, supera numericamente os softwares NeuroFurg e WEKA e em uma segunda análise, o NeuroFURG mostra-se mais eficiente para o estudo de Redes Neurais em ambiente pedagógica em relação a ferramenta WEKA.

## 6 CONCLUSÃO

Os objetivos propostos nesta pesquisa foram alcançados e para futuras pesquisas com testes de ferramentas para o estudo de Redes Neurais, sugere-se um estudo de caso onde diferentes classes de alunos possam testar o uso das ferramentas na prática e após o ensino, comparar com o teste checklist aplicado neste artigo.

Foi observado também que, mesmo o que *software* GMDH Shell DS tenha obtidos melhores resultados no teste Checklist, nenhum dos softwares testados poderiam ser considerados ruins, uma vez que cada um deles possui uma vantagem sobre os demais. O NeuroFURG apresenta funcionalidades mais educacionais, oferecendo até a geração do código fonte em várias linguagens de programação. O WEKA

disponibiliza uma vasta quantia de algoritmos, o que facilita a resolução de praticamente qualquer problema. Já no GMDH Shell DS, é possível fazer o proposto com grande facilidade, pois como visto, o software é bem intuitivo, além de ter um ambiente gráfico claro e pratico.

## 7 REFERÊNCIAS

BABELOMICS TUTORIAL. Introduction and purpose. Disponível em: <http://bioinfo.cipf.es/babelomicstutorial/predictors>. Acesso em 15 ago. 2015.

BEALE, R.; JACKSON, T. **Neural computing: an introduction**. Institute of Physics Publishing, 1990.

CASTRO, I. P. L. N.; VON ZUBEN, F. J. **Redes neurais artificiais**. Campinas: Universidade Estadual de Campinas, 2003. Disponível em: [ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia006\\_03/topico5\\_03.pdf](ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia006_03/topico5_03.pdf). Acesso em 10 set. 2015.

FISHER, R. A. The use of multiple measurements in taxonomic problems. **Annals of eugenics**, v.7, n. 2, p.179-188, 1936.

LUGER, G.; STUBBLEFIELD, W. **Artificial intelligence and the design of experts systems**. The Benjamin and Cumming Publishing Company, 1989.

MACHADO, V. P.de; LIMA, B. V. A.; ARNALDO, H. A.; ARAÚJO, S. W. I. Classificação automática dos usuários da rede social acadêmica scientia. net. In: IV CONGRESSO TECNOLÓGICO TI E TELECOM- INFOBRASIL, 2011.

MADSEN, C. A. B. C. W.; ADAMATTI, D. F. NeuroFURG: uma ferramenta de apoio ao ensino de Redes Neurais Artificiais. **Revista Brasileira de Informática na Educação**, v. 19, n. 02, p.14, 2011. Disponível em: <http://www.br-ie.org/pub/index.php/rbie/article/view/1274/1173>. Acesso em 05 set. 2015.

NIELSEN, J. **Evangelizing usability: change your strategy at the halfway point**, Jakob Nielsen's Alertbox, 2005.

OLIVEIRA, J. A., A usabilidade nos processos do ciclo de vida e na qualidade do produto de software. Manaus, 2004.

REZENDE, S. O. **Sistemas inteligentes: fundamentos e aplicações**. São Paulo: Manole, 2003.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, p. 386, 1958..

RUSSELL, S. J.; NORVIG, P. **Inteligência artificial**. 2. ed. Rio de Janeiro: Elsevier, 2004.

SILVA, Cassandra Ribeiro de O. e. **Bases pedagógicas e ergonômicas para a concepção e avaliação de produtos educacionais informatizados**. Florianópolis, 1998. Dissertação (Mestrado em Engenharia de Produção) – PPGE/UFSC.

SQUIRES, David; PREECE, Jenny. Usability and learning: evaluating the potential of educational software. **Great Britain: Computer Edu.** v. 27, n. 1, p. 15-22, 1996.