

Um modelo preditivo no diagnóstico de aprendizagem de programação**A predictive model in the programming learning diagnosis**

Recebimento dos originais: 15/03/2019

Aceitação para publicação: 04/04/2019

Lucinéia Barbosa da Costa Chagas

Mestre em Informática pela Universidade Federal do Espírito Santo

Instituição: Instituto Federal do Espírito Santo

Endereço: ES-010, Km-6,5 - Manguinhos, Serra - ES, 29173-087

E-mail: lucineia.chagas@ifes.edu.br

Jefferson Ribeiro Lima

Mestre em Informática pela Universidade Federal do Espírito Santo

Instituição: Instituto Federal do Espírito Santo

Endereço: ES-010, Km-6,5 - Manguinhos, Serra - ES, 29173-087

E-mail: jlima@ifes.edu.br

Márcia Gonçalves de Oliveira

Doutora em Engenharia Elétrica pela Universidade Federal do Espírito Santo

Instituição: Centro de Referência em Formação e em Educação a Distância

Endereço: R. Barão de Mauá, 30 - Jucutuquara, Vitória - ES, 29042-170

E-mail: clickmarcia@gmail.com

RESUMO

Inúmeras tecnologias são desenvolvidas como apoio ao processo de aprendizagem de programação. Todavia há uma grande carência de modelos eficazes nos projetos de tecnologias educacionais desta área. Para atender essa demanda, apresentamos neste artigo um modelo para criação de perfis de aprendizagem no diagnóstico de aprendizagem de programação. Esse modelo fornece os requisitos de um sistema online de monitoramento e controle dos componentes das habilidades através da criação de perfis com base nos históricos de aprendizagem. Através deste modelo visa-se alcançar progressos reais de aprendizagem na disciplina de programação de computadores.

Palavras-chaves: aprendizagem de programação, perfis de aprendizagem, previsão de desempenho.

ABSTRACT

Numerous technologies are developed to support the learning process of programming. Nevertheless, there is a great lack of effective models in the projects of educational technologies of this area. To meet this demand, we present in this article a model for creating learning profiles in the programming learning diagnosis. This model provides the requirements for an online system for monitoring and controlling skills components by creating profiles based on learning histories. Through this model we aim to achieve real learning progress in the discipline of computer programming.

Key Words: learning programming, learning profiles, performance prediction.

1 INTRODUÇÃO

A aprendizagem de programação de computadores é estudada por vários pesquisadores em todo o mundo. Todavia, mesmo com vários estudos feitos, a aprendizagem de programação continua a ser um grande desafio (Tavares, 2013).

O nível de insucesso dos estudantes nas disciplinas de programação é elevado e alvo de várias pesquisas. Dentre as dificuldades apresentadas pelos alunos no processo de programar, estão: o baixo nível de abstração, a falta de habilidades na resolução de problemas, inadequação dos métodos pedagógicos aos estilos de aprendizagem (Jenkins 2002), dificuldades na compreensão dos conceitos de programação e visões erradas sobre a atividade de programar (Souza, 2012).

Para Pascoal, *et alli* (2015) o baixo rendimento de estudantes em disciplinas de programação tem preocupado educadores e gestores das universidades, já que este fato pode ainda estar relacionado ao aumento da evasão nos cursos da área.

Diante desta problemática, este artigo apresenta um modelo para diagnosticar preditivamente o perfil dos alunos ingressantes em disciplinas de programação de computadores. Este modelo consiste na combinação de técnicas de aprendizagem de Redes Neurais e *Clustering* na previsão de desempenhos de aprendizagem.

Este artigo está estruturado da seguinte forma: a Seção 2, trás a definição do problema. Na Seção 3, é apresentado o referencial teórico. A Seção 4, aborda a metodologia aplicada no desenvolvimento do modelo de aprendizagem. Na Seção 5, apresenta o Modelo de preditivo de aprendizagem. A Seção 6, trás as considerações finais.

1.1 DEFINIÇÃO DO PROBLEMA

Silva *et al.* (2015) enfatiza que em cursos da área de computação, o ensino e aprendizagem de programação de computadores se torna essencial para uma boa formação acadêmica. Contudo, a aprendizagem de programação é considerada pela maioria dos alunos um grande desafio.

Existem alguns fatores que contribuem para um baixo rendimento e dificuldade na assimilação dos conteúdos relacionados à programação de computadores. Entre os pontos comuns destacam-se à dificuldade na compreensão dos enunciados dos problemas; incapacidade de detectar erros de lógica de programação (Gomes e Mendes, 2000).

Além da dificuldade relacionada à capacidade de abstração, há as dificuldades impostas pela sintaxe e pelas estruturas abstratas da linguagem de programação (Ribeiro, 2012).

Tavares *et.al* (2012) relatam que os alunos ingressantes trazem consigo bases diferenciadas de conhecimentos, que evidentemente influenciam nos rendimentos ao longo do curso.

Para se ajustar ao nível de exigência dos cursos superiores de computação, o aluno precisa passar por um treino intensivo na resolução de problemas, de modo a desenvolver competências para obtenção de um retorno mínimo na aprendizagem (Dijkstra,1989) e (Perkins *et al.* 1988).

Os problemas mencionados, além de dificultarem a aprendizagem de programação, reduzem a carga de exercícios, quando o ofício de programar exige extensa prática.

Hoje, poucos exercícios de programação são realizados sob a supervisão de um professor, o que se considera preocupante devido a importância desse acompanhamento (Oliveira, 2013).

Em geral as dificuldades no processo de aprendizagem de programação não são fáceis de serem solucionadas, porém podem ser amenizadas através de previsões de desempenhos baseados nos históricos de aprendizagens.

2 REFERENCIAL TEÓRICO

A aprendizagem de programação é considerada um processo em que os alunos vêm como um verdadeiro teste de vocação (Chagas, 2011).

Pesquisadores tentam entender os fatores responsáveis pelas dificuldades enfrentadas pelos alunos e analisam essas dificuldades relacionando-as com diferentes metodologias de ensino, de modo a desenvolver um melhor ambiente de aprendizagem entre professores e alunos (Pascoal, 2015).

Nesse sentido a previsão de desempenho pode ser uma grande aliada no processo de aprendizagem de programação de computadores. Uma vez que ao prever desempenhos o professor passa a ter uma visão certa do futuro do estudante no processo de ensino e de aprendizagem, o que traz ao docente a possibilidade de reorientar o aprendiz com base na antecipação dos seus resultados de aprendizagem.

A previsão de desempenho tem sido aplicada em várias modalidades de estudo. Por exemplo: na simulação de biomassa de *fitoplâncton* (Huang *et alli*, 2015).

Na avaliação de desempenho de unidades de tomada de decisão (Azadeh *et alli*, 2009), em previsões de desempenho da gravidade de drenagem assistida por vapor reservatórios heterogêneos (Amirian *et alli*, 2015).

Na previsão de desempenho de estudantes em Cursos *Online* (Brinton *et alli*, 2015), além de prever notas de exercícios conforme perfis e recomendar as atividades mais adequadas (Oliveira, 2013) dentre outros.

Neste artigo será feita uma abordagem relacionada a previsão de desempenho focada na aprendizagem de programação, no qual a previsão terá como base os históricos de aprendizagem e será realizada através de técnicas de redes neurais e *clustering*.

2.1 REDES NEURAIAS

As Redes Neurais são técnicas computacionais que apresentam um modelo matemático usado para a otimização e aprendizagem aberta, baseada em concepções inspiradas a partir do cérebro (Mirabedini, 2015).

Estudos de Amirian (2015) enfatizam que o termo Rede Neural refere-se a uma inteligência artificial usada para reconhecimento da identidade ou a aproximada relação não linear complexa entre as variáveis de entrada e saída.

Neste contexto, Berry & Linoff (1997) defendem que uma das técnicas de mineração de dados muito usada em tarefas de classificação e previsão é a modelagem por meio de redes neurais, nas quais os neurônios se comunicam através de sinapses, que é a região onde dois neurônios se comunicam através de impulsos nervosos que são transmitidos entre eles.

As aplicações de redes neurais tem sido inúmeras, podemos citar alguns exemplos, deste uso no reconhecimento de fala (Kohonen, 90), na identificação de sinais de radar (Sigillito & Hutton, 90), no uso de aplicações voltadas para o mercado financeiro (Zaremba, 90), em composições musicais (Eberhart & Dobbins, 90), na previsão de vendas (Almeida & Passari, 2006), em sistemas de recuperação de informação (Ferneda, 2011), nas previsões climáticas (Anochi, 2015), em armazenamento, categorização, formas e agrupamento de dados semelhantes (Mahdavi, *et ai* 2013) dentre outras.

A arquitetura das redes neurais se restringe ao tipo de problema que será usada, podendo ser definida por camada única ou múltiplas camadas, pelo número de nós em cada camada, pelo tipo de conexão entre os nós (*feedforward* ou *feedback*) e topologia (Haykin, 2001). A Figura 01 apresenta o modelo de neurônio artificial que é uma simplificação do modelo apresentado por Haykin (2001).

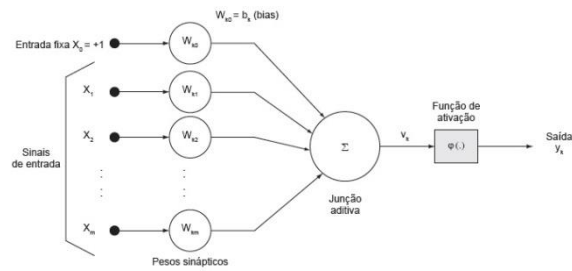


Figura 01 – Modelo não linear, Haykin (2001)

O modelo apresentado na Figura 01 é composto por três elementos: um conjunto de conexões de entrada (x_1, x_2, \dots, x_n), caracterizadas por pesos ($W_{k0}, W_{k1}, \dots, W_{kn}$), um somador para acumular os sinais de entrada, uma função de ativação que limita o intervalo permissível de amplitude do sinal de saída (y) a um valor fixo.

O comportamento das conexões entre os neurônios é simulado por meio de seus pesos. O efeito de um sinal proveniente de um outro neurônio é determinado pela multiplicação do valor da intensidade do sinal recebido pelo peso da conexão correspondente.

Em seguida é efetuada a soma dos valores de todas as conexões, e o valor resultante é enviado para a função de ativação, que define a saída (y) do neurônio. Combinando diversos neurônios, é formado uma rede neural artificial.

Dentre os modelos de redes neurais destacamos: o *Perceptron* e *Perceptron* multicamadas. Em que o *Perceptron* possui a capacidade de resolver problemas de forma linear separável por meio do processo de treinamento.

Neste caso, o algoritmo aprende a classificar as entradas em apenas dois grupos diferentes.

Já o *Perceptron* multicamadas é semelhante ao primeiro, com a diferença de possuir várias camadas de neurônios. Sendo assim, onde não existir a possibilidade de uma reta única, é gerado mais de uma reta classificadora.

Para o desenvolvimento desta pesquisa optamos pelo uso do modelo *Perceptron* multicamadas juntamente com um algoritmo de retropropagação. Optamos por este modelo devido ao fato de possuir uma probabilidade menor de erros se comparado aos demais modelos existentes.

2.2 CLUSTERING

O *clustering* é uma técnica de mineração de dados que usa a abordagem de aprendizagem não-supervisionada para agrupamento de padrões considerando as características semelhantes desses padrões (Oliveira, 2013).

Os resultados apresentados nos trabalhos de Pimentel *et al* (2003) confirmam que técnicas de *clustering* são bastante úteis para a formação de grupos homogêneos de aprendizes.

O objetivo do *clustering* é formar grupos caracterizados por alta homogeneidade entre padrões de um mesmo grupo e entre padrões de grupos diferentes levando em consideração os diferentes perfis de conhecimento.

A Figura 02, mostra um processo de *clustering* em que, as figuras se auto-organizam formando grupos a partir de sua proximidade.

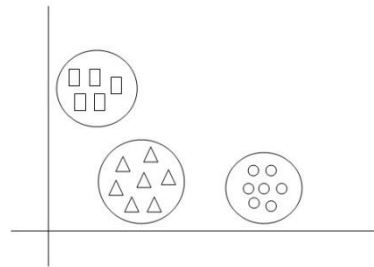


Figura 02. Formação de grupos de proximidade através da técnica de *cluster*. Fonte: Autor

Jain, *et al* (1999) afirmam que uma atividade padrão de *clustering* envolve os cinco passos destacados na tabela 01.

Tabela 01 – Passos para atividade padrão *clustering*

Passos	Descrição
Representação de padrões	Na representação de padrões, pode haver seleção e extração de características. A seleção consiste na identificação de um subconjunto dos atributos de um padrão para descrevê-lo. Já extração consiste em realizar transformações nos atributos de um padrão para melhor descrevê-lo.
Definição de medidas de similaridade	A definição de medidas de similaridade são funções de distância definidas entre pares de padrões, como: <i>coseno</i> , <i>coeficiente de Jaccard</i> , <i>coeficiente de correlação</i> e <i>distância euclidiana</i> .
	Este passo pode ser realizado por diferentes métodos de <i>clustering</i> . Sendo

Agrupamento	as abordagens hierárquica e particional mais comuns.
Abstração de dados, se necessário	A abstração de dados é o processo de extrair uma representação compacta de um conjunto de dados direcionada para análise automática ou para o usuário.
Avaliação da saída, se necessário	Geralmente se recorre a critérios de otimização, o fato é que os <i>clusters</i> são produzidos a partir dos dados de entrada. Desse modo, o processo de <i>clustering</i> não pode ser validado como eficiente ou não caso os padrões de entrada em sua essência não se agrupem.

A principal vantagem do *clustering* sobre classificação está no fato de ser adaptável a destacar alterações e características úteis que ajudam distinguir grupos diferentes (Oliveira, 2013).

Quanto a função de mineração de dados, a análise de *clustering* serve para visualizar a distribuição de dados e observar as características de cada agrupamento.

2.3 SOFTWARE WEKA - WAIKATO ENVIRONMENT FOR KNOWLEDGE ANALYSIS

O software *WEKA* é uma ferramenta de Descoberta de Conhecimento em Bancos de Dados (*Knowledge Discovery in Databases - KDD*) que contempla uma série de algoritmos de preparação de dados, de aprendizagem de máquina e de validação de resultados (Silva, 2004).

Desenvolvido na Universidade de *Waikato* na Nova Zelândia, o *WEKA* foi implementado pela primeira vez em 1997.

Escrito em linguagem Java, contém uma *GUI* para interagir com arquivos de dados e produzir resultados visuais, além de uma *API* geral, que torna possível incorporá-lo em aplicativos para desenvolver tarefas de mineração de dados automatizadas no lado do servidor. Markov & Russell (2006) apontam algumas características relevantes deste software, como:

- Ser relativamente de fácil usabilidade;
- Possuir diversos algoritmos para *data mining*, *web mining* e *machine learning*;
- Possuir código aberto;
- Estar disponível gratuitamente na *Web*;
- Estar atualizado;
- Proporcionar recursos flexíveis para experimentos;

Para Silva (2004), o sistema possui uma interface gráfica amigável e seus algoritmos fornecem relatórios com dados analíticos e estatísticos do domínio minerado.

3 METODOLOGIA

Neste trabalho as redes neurais são usadas com a finalidade de mapear os perfis de aprendizagem. Neste caso, são usadas as redes neurais multicamadas *Perceptron*, juntamente com o algoritmo de retropropagação, usado no treinamento de redes neurais multicamadas com uma ou mais camadas escondidas.

Este algoritmo consiste em dois passos de computação: No primeiro passo, o processamento é direto no qual uma entrada é aplicada à rede neural e seu efeito é propagado pela rede, camada a camada.

O segundo passo é o processamento reverso em que, um sinal de erro calculado na saída da rede é propagado no sentido reverso, camada a camada, e ao final deste processo os pesos são ajustados de acordo com uma regra de correção de erro.

Em seguida com os resultados produzidos no mapeamento dos perfis é aplicado a análise de *clustering*, em que as características dos padrões são analisadas para a partir delas descobrir possíveis classes de padrões (Oliveira, 2013).

Por fim é aplicado o algoritmo de classificação *Bisecting K-means*, onde reúne os padrões por classes já conhecidas. O algoritmo *Bisecting K-means* é uma variação do algoritmo *K-means*, no qual começa com um simples *cluster* e continuamente seleciona um *cluster* para dividir em dois *sub-clusters* até alcançar o número *K* de *clusters* desejados (Oliveira, *et al* 2014).

4 MODELO PREDITIVO PARA DIAGNÓSTICO DE APRENDIZAGEM DE PROGRAMAÇÃO

A ideia do modelo preditivo para diagnóstico de aprendizagem de programação está em prever desempenhos de aprendizagem com base nos históricos dos estudantes. Para atingir este objetivo são usadas técnicas de Redes Neurais e *Clustering*.

O modelo preditivo possui duas etapas em que, na primeira, é usado um conjunto de exercícios desenvolvidos pelos discentes. Esses exercícios servem como entrada no *cluster* para selecionar amostras do treino do modelo.

De cada *cluster*, são selecionadas 2/3 de amostras aleatórias para gerar modelos para prever *scores* indicadores de níveis de aprendizagem. Após, os *scores* dos demais 1/3 das amostras são preditos e uma classe de nível de aprendizagem é associada a cada uma delas.

Em seguida estas amostras são inseridas em um banco de amostras juntamente com um exercício de prova onde são pontuadas pelo professor para posteriormente serem usadas para o treinamento da rede neural.

Na segunda etapa do Modelo Preditivo, o módulo responsável por mapear as classes de notas previstas recebe como entrada os componentes de habilidades (*CH*).

Os componentes de habilidade são características de perfis de alunos representados em vetores. Estas características estão relacionadas à frequência de ocorrência de palavras reservadas, como: símbolos, operadores e funções da linguagem ou indicadores de funcionamento como a compilação e a execução de programas. Conforme Figura 03.

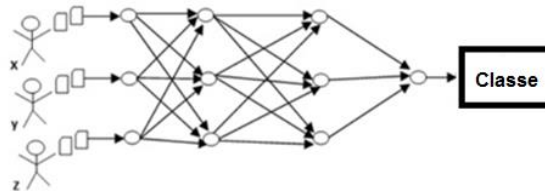


Figura 03 – Técnica de Rede Neural aplicada ao modelo Preditivo. Fonte: do Autor

Esses *CHs* são representados por pesos, em seguida é feita a multiplicação do valor da intensidade do sinal recebido pelo peso da conexão correspondente.

Após este procedimento é efetuado a soma dos valores de todas as conexões, e o valor resultante é enviado para a função de ativação, que define a saída do neurônio. Esta saída é chamada de *classe* que corresponde à nota prevista do discente. A Figura 04 apresenta a linha do tempo do Modelo Preditivo.

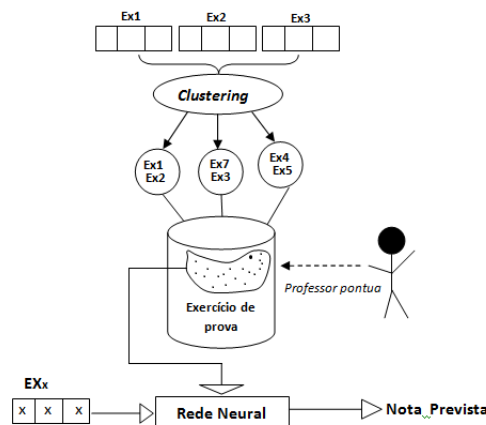


Figura 04 – Linha do tempo do Modelo Preditivo. Fonte: do Autor

Para a aplicação dos experimentos foi desenvolvido uma sistema de aprendizagem de programação. Este sistema possui dentre outras funcionalidades uma interface para interação do aluno e outra para o professor.

Na visão do aluno é possível escolher a tarefa a ser executada e fazer o envio dos códigos desenvolvidos. A Figura 05 apresenta o *layout* do sistema.

Figura 05 – SAProg – Sistema para Acompanhamento de Programação - visão do Aluno.

Fonte: do Autor

O sistema recebe os códigos desenvolvidos pelos alunos e em seguida esses exercícios vão para uma análise *Clustering* servindo como entrada para seleção de amostras do treino do modelo.

Em seguida as amostras selecionadas são inseridas em um banco de amostras juntamente com o exercício de prova para serem pontuados pelo professor e posteriormente servir como treino para as redes neurais.

Na visão do professor, todo este procedimento feito no núcleo do sistema fica oculto. O docente tem acesso apenas aos resultados fornecidos por ele, como mostra a Figura 06.

Perfil 1	Perfil 2	Perfil 3
Aluno C	Aluno A	Aluno B
Aluno E		Aluno D
Aluno F		
Aluno G		
Regular	Bom	Ótimo

Clique sobre o nome do aluno para acompanhar o rendimento individual.

Figura 06 – SAProg – Sistema para Acompanhamento de Programação – visão do Professor.

Fonte: do Autor

Na tela fornecida pelo Sistema de acompanhamento de aprendizagem, o professor tem a possibilidade de escolher a tarefa cadastrada. Após escolha são fornecidos os resultados do processamento, resultando as notas dos alunos.

Por meio do *feedback* fornecido pelo sistema, o docente tem a possibilidade de elaborar planos pedagógicos individualizados para um melhor acompanhamento e aplicação de metodologias mais especializadas.

O que facilita o envolvimento e aptidão de cada grupo para o ensino de programação, tornando o processo de aprendizagem do conteúdo mais prazeroso para os discentes.

Além disso, o usuário do sistema que possui perfil de *Administrador* pode cadastrar novas tarefas, definir outros perfis e atribuir os seus respectivos pesos de acordo com especificidade do processo para mapeamento e análise dos grupos que estudam alguma linguagem de programação.

Os dados cadastrados pelo *Administrador* do sistema podem ser aplicados no diagnóstico da aprendizagem dos discentes por meio do processamento interno do algoritmo do modelo preditivo para posterior mapeamento e auxílio direcionado aos alunos.

5 CONSIDERAÇÕES FINAIS

O objetivo do modelo preditivo apresentado está em estender as práticas do professor através de previsões que possibilite a promoção de uma aprendizagem real de programação.

As previsões de desempenhos com base nos históricos de aprendizagem são previstas com uso de técnicas de redes neurais multicamadas, *clustering* e algoritmos de classificação.

Pretende-se que a aprendizagem de programação possa ser melhorada com a aplicação do modelo apresentado pelo fato de as estratégias de previsão de desempenhos possibilitarem o controle do processo de aprendizagem em nível de habilidades.

Para o controle de aprendizagem, o estudante é mapeado em habilidades e nos desempenhos alcançados no domínio de conteúdos.

Com um diagnóstico das lacunas de aprendizagem, o professor tem a possibilidade de intervir nas variáveis de aprendizagem que apresentam deficiências realimentando-as até que elas apresentem êxitos.

A princípio, os testes serão aplicados em um laboratório de informática com aproximadamente 40 alunos do curso técnico do primeiro módulo, em uma instituição de ensino federal localizado no estado do Espírito Santo, Brasil.

A escolha do curso para realização do primeiro experimento foi motivado pelo perfil do aprendiz, que neste caso, normalmente não possui nenhum tipo de conhecimento prévio em programação de computadores (Oliveira, 2013).

Em cada aula prática o aluno do primeiro módulo do curso técnico de informática irá receber uma questão relacionada ao ensino teórico recebido.

Deste modo, o discente tentará resolver o problema de cunho computacional utilizando uma linguagem de programação definida pelo docente.

No término de cada aula prática, o aluno deverá ser autenticado no sistema *SAProg* para enviar o código desenvolvido na aula. Os testes serão previamente elaborados com a perspectiva de reforçar a teoria e a prática com relação às questões relacionadas ao ensino de programação.

A cada questão prática solucionada pelo discente, o novo experimento terá seu nível de complexidade gradualmente amplificado. No segundo momento, após obtenção do primeiro resultado com os alunos do ensino técnico, pretendemos ampliar a área de testes do modelo proposto utilizando como um novo cenário o curso superior de informática.

Para finalizar, destacamos que após aplicação e geração dos resultados do primeiro experimento, por meio do modelo preditivo de diagnóstico de aprendizagem, serão publicados os laudos que poderão certificar a eficácia do modelo apresentado.

REFERÊNCIAS

ALMEIDA, F. C. & PASSARI, A. F. L.: Previsão de vendas no varejo por meio de redes neurais. *Revista de Administração- RA USP-* 2006.

ANOCHI, J. A.: Previsão climática de precipitação por redes neurais autoconfiguradas. Tese de doutorado – INEP, 2015.

AMIRIAN, E., LENUNG, J. Y., ZANON, S., DZURMAN, P.: *Integrated cluster analysis and artificial neural network modeling for steam-assisted gravity drainage performance prediction in heterogeneous reservoirs. In: Expert Systems with Applications – 2015.*

AZADEH A., Saberi, M., Anvari, M., Mohamadi, M.: *An integrated artificial neural network-genetic algorithm clustering ensemble for performance assessment of decision making units. Springer Science+Business Media – 2009.*

BRINTON, C. G.; CHIAN, M.: *MOOC Performance Prediction via Clickstream Data and Social Learning Networks*. In: *IEEE Xplorer Digital Library*, 2015.

BERRY, Michael J.A.; LINOFF, Gordon. *Data mining techniques for marketing, sales and customer support*. 1st ed. [S.l.]: John Wiley & Sons, 1997.

EBERHART, R. & DOBBINS, R. Case Study V: *Making Music*. In: *Neural Networks PC Tools - A Practical Guide*. Eberhart, R. & Dobbins, R. Academic Press, San Diego, 1990.

DA SILVA, I. C., FONSECA, L. C. C., & JESUS, R. D. Um Sistema Tutor Inteligente para o Ensino no Domínio de Lógica de Programação. TISE 2015.

FERNEDA, E. Redes neurais e sua aplicação em sistemas de recuperação de informação. In: *Scientific Eletronic Library Online*, 2011.

HAYKIN, S. *Redes neurais: princípios e prática*. 2. ed. Porto Alegre: Bookman, 2001.

HUANG, J.; Gao, J.; Zhang, Y.: *Combination of artificial neural network and clustering techniques for predicting phytoplankton biomass of Lake Poyang, China*. Springer-2015.

JENKINS, T. (2002). *On the difficulty of learning to program*. In *Proceedings of 3rd Annual LTSN_ICS Conference (Loughborough University, United Kingdom, August 27-29, 2002)*. The Higher Education Academy, p.53-58.

KOHONEN, T. *The Self-Organizing Map*. *Proceedings of the IEEE*, V.78, n. 9, Sep. 1990.

M. Mahdavi, M. Taheri, F. Lotfi, "Data mining application in artificial neural networks", *The 8th Symposium on Advances in Sciences and Technology, Mashhad, Iran, 2013*.

MACCULLOCH, W. S.; PITTS, W. *A logical calculus of the ideas immanent in nervous activity*, *Bulletin of Mathematical Biophysics*, [S.l.], v. 5, p. 115-133, 1943.

MARKOV, Z.; RUSSELL, I. *An Introduction to the WEKA Data Mining System*. *Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education*. P. 367 - 368. Bologna, Italy, 2006.

MIRABEDINI, S. J. ; SAFA, H. M.; DELAVAR, Z.: *Artificial neural networks applications in data clustering*. In: *DU Journal, Humanities and Social Sciences*, 2015.

OLIVEIRA, M. G. Núcleos de avaliações diagnóstica e formativa para regulação da aprendizagem de programação. Tese de doutorado-2013.

OLIVEIRA, M. G. MONROY, N. A. J.; ZANDONADE, E.; OLIVEIRA, E.: Análise de Componentes Latentes da Aprendizagem de Programação para Mapeamento e classificação de Perfis. In: III Congresso Brasileiro de Informática na Educação - CBIE 2014.

PASCOAL, T. A.; BRITO, D. M.; RÊGO, T. G: Uma abordagem para a previsão de desempenho de alunos de Computação em disciplinas de programação. In: Nuevas Ideas en Informática Educativa TISE 2015.

PIMENTEL, Edson P.; FRANÇA, Vilma F.; OMAR, Nizam. (2003) A identificação de grupos de aprendizes no ensino presencial utilizando técnicas de clusterização. XIV Simpósio Brasileiro de Informática na Educação.

ROSENBLATT, F. *The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review, Washington, v. 65, n. 6, p. 386-408, 1958.*

SIGILLITO, V. & HUTTON, L. *Case Study II: Radar Signal Processing. In: Neural Networks PC Tools - A Practical Guide. Eberhart, R. & Dobbins, R. Academic Press, San Diego, 1990.*

SILVA, M. P. S.: Mineração de Dados: Conceitos, Aplicações e Experimentos com Weka. Livro da Escola Regional de Informática Rio de Janeiro - 2004.

SOUZA, D. M.; MALDONADO, J.C.; BARBOSA, E. F: Aspectos de Desenvolvimento e Evolução de um Ambiente de Apoio ao Ensino de Programação e Teste de Software. Anais do 23º Simpósio Brasileiro de Informática na Educação (SBIE 2012) - Rio de Janeiro – 2012.

SUH, S. C. (2012). *Practical applications of data mining. USA: Jones Bartlett Learning.* Takagi, H., & Hayashi, I. (1991). *NN-driven fuzzy reasoning. International Journal of Approximate Reasoning (Special Issues of IIZUKA '88), 5(3), 191–212.*

TAVARES, O. L.; MENEZES, C.S.; ARAGÓN, R.; COSTA, L. B.: Uma arquitetura pedagógica auxiliada por tecnologias para ensino e aprendizagem de programação. In XXXIII Congresso da sociedade brasileira de computação - CSBC 3013.

ZAREMBA, T. Case Study III: *Technology in Search of a Buck*. In: *Neural Networks PC Tools - A Practical Guide*. Eberhart, R. & Dobbins, R. Academic Press, San Diego, 1990.