

Um Mapeamento Sistemático sobre Métodos de Identificação Preditiva de Alunos com Risco de Reprovação em Educação de Computação

Lucas R. Costa , Caique L. Souza ,
Ana Carolina G. Inocêncio , Esdras L. Bispo Jr.

¹Instituto de Ciências Exatas – Universidade Federal de Jataí (UFJ)
75800-000 – Jataí – GO – Brasil

lucas.costa67@yahoo.com.br, caiquel2@hotmail.com,

{anacarolinaufg, bispojr}@ufg.br

Abstract. *The student assessment, formative or summative, is a topic of interest in Computer Science Education. One of the important functions of the assessment is to estimate the risk of some students do not obtain a satisfactory performance in the discipline. This mapping is intended to list the difficulties of predicting the final performance of students in of the course of Computer Science, in the beginning of the discipline, as well as proposed methods for such. The mapping has special emphasis on identifying methods that use data from teaching methodologies to adopt continuous, formative evaluation, like the Peer Instruction. In total, 81 papers were found. When applying the inclusion and exclusion criteria, it was selected 33 papers for a perusal, which led the choice of 18 papers for this work. During the mapping, it was realized that it is possible to predict, with a certain level of accuracy, students with risk of rejection and that, once avoiding her, to reduce the chance of evasion of these students.*

Resumo. *A avaliação do discente, sendo ela formativa ou somativa, é um tópico de interesse na educação de computação. Uma das funções importantes da avaliação é estimar o risco de alguns alunos não obterem um desempenho satisfatório na disciplina. Este mapeamento tem a intenção de elencar as dificuldades de se prognosticar o desempenho final dos alunos do curso de Ciência da Computação, ainda no início da disciplina, bem como métodos propostos para tal. No total, foram encontrados 81 trabalhos. Ao aplicar os critérios de inclusão e exclusão foram selecionados 33 trabalhos para uma leitura minuciosa, o que levou a escolha de 18 artigos para este artigo. Durante o mapeamento, percebeu-se que é possível prever, com certo nível de precisão, os alunos com risco de reprovação e que, uma vez evitando-a, diminui-se a chance de evasão destes alunos do curso.*

1. Introdução

Um excelente ponto de partida para descrever a Pesquisa em Educação de Computação (PEC) são as duas áreas de sua junção: a Educação e a Ciência da Computação. Pode-se definir, como um dos objetivos principais da PEC, o aperfeiçoamento do processo de ensino e aprendizagem da Computação como ciência [Holmboe et al. 2001]. Existem

muitos desafios abertos na PEC [Robins 2015]. Um deles está associado ao alto índice de evasão registrado em cursos superiores de Computação.

A aprendizagem da Computação, como Ciência, pode apresentar dificuldades para o aprendiz iniciante, assim como em outras áreas de exatas [Blando 2015]. Segundo dados do Censo da Educação Superior de 2017 [INEP 2018], publicado pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) do Ministério da Educação (MEC), a taxa de evasão do Bacharelado em Ciência da Computação alcançou cerca de 60,2%, enquanto a taxa geral ficou em torno dos 26%, mostrando que o problema está se agravando no curso.

Muitas vezes os ingressantes não conseguem acompanhar o fluxo de estudos, têm baixo desempenho e assim acabam evadindo ou reprovando nas matérias introdutórias do curso. Com taxas de evasão tão altas, um dos desafios constantes da PEC é identificação prévia de estudantes em risco de reprovação ou evasão [Porter and Zingaro 2014].

Muitos métodos de identificação já foram propostos na literatura, como o modelo de ARL (Análise de Regressão Linear) utilizando dados coletados da IpC (Instrução pelos Colegas) para previsão do desempenho do exame final [Liao et al. 2016]. Uma outra proposta é o modelo PreSS que usa fatores comparativos - como a eficiência de programação, a habilidade matemática e as horas dedicadas em exercícios - para auxiliar a identificação do alunos em risco de reprovação [Quille and Bergin 2018]. Existem até ferramentas capazes de realizarem capturas de telas, que exibem código-fonte, para auxiliar na identificação das dificuldades apresentadas pelos alunos usando aprendizado de máquina [Ahadi 2016]. Entretanto, ainda existe uma lacuna no que diz respeito a trabalhos secundários na área, como por exemplo, mapeamento sistemático.

Este trabalho tem como propósito realizar um mapeamento sistemático sobre os métodos de identificação preditiva de alunos com risco de reprovação em Educação de Computação. O objetivo deste mapeamento é possibilitar pesquisadores a identificar os principais métodos propostos na literatura, contrastando-os entre si ou com possíveis novos métodos a serem propostos no futuro. Este mapeamento tem como ênfase especial a identificação de métodos que utilizem dados oriundos de metodologias de ensino que adotem avaliação formativa contínua [Aranha 2017], como a Instrução pelos Colegas [Crouch and Mazur 2001].

Além disso, este mapeamento permitirá que novas pesquisas provejam para o professor ou gestor educacional um leque de ferramentas que os auxiliem na tomada de decisão em vistas às ações político-pedagógicas que enfrentem o problema do alto índice de reprovação e de possíveis casos futuros de evasão.

O restante do trabalho é dividido como se segue. A Seção 2 apresenta a descrição metodológica do como foi realizado o mapeamento sistemático. A Seção 3 apresenta uma discussão em relação aos resultados encontrados pelos artigos levantados neste mapeamento. E, por fim, na Seção 4, as considerações finais são apresentadas.

2. Metodologia

Para a produção deste mapeamento sistemático, nos baseamos no modelo utilizado por [Ramos and Brasil 2012]. Na Seção 2.1. é descrito o problema, através da pergunta central da pesquisa. A Seção seguinte apresenta o foco do mapeamento, com a descrição

do objetivo esperado. A Seção 2.3. apresenta as questões que nortearam a pesquisa, em torno da questão principal. A Seção 2.4 descreve as palavras-chaves e as strings de busca, tanto em português quanto em inglês, utilizadas para a pesquisa. E, por fim, a Seção 2.5. descreve os critérios utilizados para a inclusão ou exclusão dos trabalhos que fossem relevantes para esta pesquisa, as máquinas de busca, onde foram aplicados as strings de busca e os recortes - de idioma e período de publicação. A Seção relata também os resultados encontrados, as quantidades de artigos escolhidos em cada etapa do processo de crivo dos trabalhos.

2.1. Descrição do Problema

Como identificar alunos em situação de risco de reprovação em uma disciplina na Ciência da Computação?

2.2. Objetivo

O objetivo deste mapeamento sistemático é fazer uma revisão, seleção e organização dos artigos que possam identificar os alunos em situação de risco, analisando os métodos utilizados pelos outros autores para solucionar o problema supracitado. Há uma ênfase especial neste mapeamento em relação à metodologias que utilizam avaliação formativa contínua, tais como a Instrução pelos Colegas.

2.3. Questões de Pesquisa

Para nortear esta pesquisa, foram definidas questões que levantem pontos relevantes a serem estudados.

- **Questão 1:** Quais métodos existem para identificar risco de reprovação em disciplinas?
- **Questão 2:** Quais os indícios de que um aluno pode ser reprovado, baseado em dados gerados pela avaliação formativa?
- **Questão 3:** Quais as medidas que existem para impedir esta reprovação?

2.4. Palavras-chave e strings de busca

As palavras-chave que compõem a busca são (em português e inglês, respectivamente): Educação de Computação, Ensino de Computação, Introdução a Computação, Avaliação, Estimativa, Previsão, Inteligência Artificial, Mineração de Dados, Aprendizado em Profundidade, Redes Neurais Artificiais, RNA, *Computer Education*, *CSI*, *Assessment*, *Prediction*, *Artificial Intelligence*, *Data Mining*, *Clicker*, *Deep Learning*, *Artificial Neural Network*, *ANN*.

A string de busca foi gerada a partir da combinação das palavras chave e foram divididas pelos idiomas da seguinte forma:

Português: (“Educação de computação” OR “ensino de computação” AND “Introdução a Computação” OR “Avaliação”) AND (“estimativa” OR “previsão” OR “Inteligência Artificial” OR “mineração de dados” OR “Clicker” OR “Aprendizado em Profundidade” OR “Redes neurais” OR “RNA”)

Inglês: (“*computer education*” AND “*csi*” OR “*assessment*”) AND (“*prediction*” OR “*data mining*” OR “*artificial intelligence*” OR “*clicker*” OR “*deep learning*” OR “*Artificial neural network*” OR “*ANN*”))

2.5. Método utilizado para pesquisa

Os seguintes critérios de inclusão foram definidos e aplicados na seleção dos trabalhos relevantes: (i) o trabalho deve estar escrito em inglês ou português; (ii) a versão completa do trabalho estar disponível na internet; (iii) apresentar necessidade de identificar os alunos em situação de riscos; (iv) apresentar os métodos utilizados para identificar esses alunos. Como critério de exclusão, foram removidos aqueles trabalhos que são réplicas ou projetos de pesquisas iniciais, priorizando o trabalho mais completo (Geralmente, o mais recente).

Aplicou-se o método de pesquisa preestabelecido para a identificação de potenciais trabalhos relacionados ao tema deste mapeamento. As máquinas de busca utilizadas para a pesquisa foram: *ACM Digital Library*¹ e *Google Scholar*². Os idiomas definidos para busca e seleção dos artigos foram português e inglês, no período de 2013 a 2018. Foram recuperados, no total, 67 trabalhos em inglês e 14 em português.

Foi realizada uma triagem, que consistiu na leitura do título, do resumo (*abstract*) e das palavras-chave (*keywords*) dos trabalhos previamente recuperados, aplicando os critérios de inclusão e exclusão. 33 trabalhos foram pré-selecionados; 3 dos 48 rejeitados nessa etapa, se tratavam apenas do resumo. A maioria foi rejeitada - 40 artigos, pois não apresentavam métodos para identificar os alunos com possibilidade de reprovação e muitos deles tratavam apenas de métodos e ferramentas para auxílio do ensino.

A partir do conjunto preliminar de trabalhos, realizou-se uma leitura minuciosa dos textos completos dos mesmos. Nesta etapa, os critérios de inclusão e exclusão foram reaplicados. Dos 33 trabalhos selecionados, 18 foram escolhidos. A maioria - 52 artigos dos 63 rejeitados no total - foi rejeitada pois não apresentavam a necessidade de identificar os alunos em situação de riscos e, conseqüentemente, não apresentavam métodos para tal.

3. Resultados e Discussões

Ao ler os artigos selecionados, foram encontrados diversos métodos na tentativa de identificar os alunos com risco de reprovação, e a maioria - 11 artigos, dentre os 18 escolhidos - dos métodos encontrados utilizavam técnicas estatísticas na análise dos dados. Outros trabalhos utilizaram como técnica preditiva a aprendizagem de máquina (*machine learning*) e outras técnicas de Inteligência Artificial.

Nesta Seção, será apresentado uma discussão em torno das questões estabelecidas para nortear o artigo. Na primeira questão (Seção 3.1), serão apresentadas as técnicas preditivas usadas nos trabalhos correlatos selecionados no mapeamento. A segunda questão (Seção 3.2) trata dos indícios de reprovação que podem ser encontrados na avaliação do aluno. E, por fim, a terceira questão (Seção 3.3) traz à tona a discussão sobre o que pode ser feito para evitar estas situações.

3.1. Questão 1: Quais métodos existem para identificar risco de reprovação em disciplinas?

A Tabela 1 apresenta a relação dos artigos estudados neste mapeamento e os métodos utilizados por cada um, no esforço de realizar a predição de desempenho dos alunos. A Tabela também apresenta a quantidade de artigos associada a cada método.

¹<https://dl.acm.org/>

²<https://scholar.google.com.br>

Tabela 1. Métodos de identificação de risco de reprovação utilizado em cada trabalho selecionado no mapeamento.

Artigos / Métodos	ARL	CPMP / Phi	Aprend. de Máquina	Árvore de Decisão	WEKA	Outros
[Danielsiek 2016]	×					
[Liao et al. 2016]	×					
[Quille and Bergin 2018]	×					
[Watson et al. 2014]	×					×
[Estey et al. 2017]						×
[Lambert 2015]		×				
[Haden et al. 2017]		×				
[Porter and Zingaro 2014]		×				
[Lishinski et al. 2016]		×				
[Ahadi et al. 2016]		×				
[Ahadi 2016]			×			
[Castro-Wunsch et al. 2017]			×			
[Gotardo et al. 2013]			×			
[Brito et al. 2014]					×	
[Amorim et al. 2008]					×	
[Detoni et al. 2014]					×	
[Silva et al. 2014]				×		
[Ferreira 2016]				×		
TOTAL	4	5	3	2	3	2

Em [Liao et al. 2016], os dados foram extraídos ao utilizar o método IpC (Instrução pelos colegas)³ e, com esses dados, construiu-se um **Modelo de ARL** (Análise de Regressão Linear) para examiná-los, o qual criado através dos principais componentes, escolhido pelo PCA (*Principal Component Analysis*), prevendo a nota que seria obtida no exame final. Em [Porter and Zingaro 2014], também foi utilizado o IpC na coleta de dados, porém os autores utilizaram a **CPMP (Correlação de Produto-Momento de Pearson)** para analisar os dados, e assim verificar o quão relacionado está o desempenho do aluno ao decorrer do curso com o sucesso obtido no final do mesmo, e concluiu-se que há correlação entre o resultado obtido entre a 3^a e a 4^a semana com o resultado final do aluno.

Em [Lishinski et al. 2016], foram utilizadas tarefas de múltipla escolha, em que cada item possuía ponderações diferentes para medir a capacidade dos alunos na resolução de problemas, podendo assim calcular a CPMP entre os resultados obtidos em aula e o resultado final. Uma abordagem mais ampla é utilizada em [Lambert 2015]. Ainda adota-se a CPMP, porém com alguns fatores que podem levar o aluno à reprovação (entre eles, a capacidade matemática e o conhecimento prévio em programação). A CPMP foi utilizada em [Haden et al. 2017] para que se possa medir a afetividade do aluno em relação ao am-

³Também conhecido como *Peer Instruction* [Müller et al. 2017].

biente e à influência no resultado obtido no CS1 (*Computer Science I* - Introdução). Outros autores utilizaram a CPMP em conjunto com a **Coefficiente Phi** [Ahadi et al. 2016], de forma que conseguiram identificar relações existentes entre o número de tentativas na resolução dos problemas de programação e o resultado final obtido.

Em [Danielsiek 2016], após a extração dos dados, foi adotada a ARL e verificada a relação entre os resultados obtidos durante as aulas e os resultados finais da disciplina. Uma outra abordagem utilizou uma ARL mesclada ao teste de normalidade **Shapiro-Wilks**, testando o quão preditor um conjunto de 50 variáveis pode ser ao tentar identificar os alunos em situação de risco de reprovação, mostrando qual variável tem maior desempenho em prever o resultado final do aluno [Watson et al. 2014]. Já em [Quille and Bergin 2018], foi apresentado o **Modelo PreSS**, criado há 20 anos, que teria a capacidade de identificar os alunos em situação de risco de reprovação (também utilizando ARL).

Alguns trabalhos utilizaram outros métodos como, por exemplo, **Aprendizado de Máquina**. Em [Ahadi 2016], utilizou-se uma ferramenta capaz de capturar *snapshots* dos códigos-fonte produzidos pelos alunos, em um curso introdutório de programação, de modo que o algoritmo identificava padrões de alunos com dificuldades. Já em [Castro-Wunsch et al. 2017], com o auxílio de uma Rede Neural Artificial (RNA), foi realizada a identificação dos alunos que estão em risco de reprovação. Uma outra abordagem utilizava uma ferramenta que coletava o número de submissões feitas e a quantidade de dicas usadas pelos alunos para que o modelo criado pudesse realizar o cálculo das **Métricas de Trajetória e de Linha de Base** para a identificação dos alunos que necessitavam de auxílio [Estey et al. 2017].

Outro trabalho apresentou uma abordagem usando algoritmos de aprendizado de máquina [Gotardo et al. 2013]. Os algoritmos foram acoplados para integrar diferentes técnicas de aprendizado e mineração de dados e explorar um conjunto de dados educacionais. Os algoritmos foram *Naive Bayes* e J48.

Alguns trabalhos utilizaram, como técnica de identificação de alunos em situação de risco de reprovação, a **Árvore de Decisão**. Um destes propõe um algoritmo, usando a técnica de árvore de decisão, que identificava os alunos em risco e gerava relatórios para acompanhamento, utilizando o ambiente virtual de aprendizagem (AVA) *Moodle* [Silva et al. 2014]. O trabalho apresentou uma precisão de 72%.

Outro trabalho, que utilizou a árvore de decisão, propôs um modelo para predição de grupos de risco [Ferreira 2016]. A técnica de árvore de decisão foi utilizada para possibilitar um diferencial quanto à possibilidade de interceptação dos dados gerados pelo uso dos métodos de predição, pois outros métodos, como Redes Neurais Artificiais (RNA), possuem como deficiência justamente a dificuldade de identificar as causas que levam aos resultados das predições.

Entre os 18 trabalhos correlatos selecionados, 3 utilizaram o **Ambiente WEKA**. O WEKA é um pacote de software que acopla vários algoritmos de Inteligência Artificial. Os trabalhos escolhidos que utilizaram a ferramenta analisaram a atuação de alguns dos principais algoritmos para a identificação de grupos de risco.

Um destes trabalhos visava identificar os alunos que necessitavam de apoio didático no início do curso, utilizando Mineração de Dados Educacionais para evitar a

evasão do curso [Brito et al. 2014]. O trabalho avalia a relação entre as notas de ingresso do aluno e o seu desempenho no primeiro período do curso. Foram utilizados os algoritmos *Naive Bayes*, *IBk*, *SMO*, *Random Florest* e *MultiPerceptron*. O trabalho obteve precisão superior a 70%.

Outro trabalho ao utilizar o WEKA apresentou as principais fases para implementação de um sistema de previsão que testa a acurácia de três classificadores amplamente utilizados e mostra as estatísticas referentes a evasão em cada curso [Amorim et al. 2008]. Os algoritmos utilizados foram redes bayesianas, *SMO* e *J48* (os 3 principais algoritmos do WEKA).

E, por fim, uma abordagem que desenvolveu e testou as técnicas e modelos de aprendizagem de máquina no AVA *Moodle* [Detoni et al. 2014]. OS modelos foram gerados e avaliados usando WEKA com os seguintes algoritmos: rede bayesiana, rede neural, *J48* e *Random Florest*. A acurácia média alcançada foi superior a 90%. A precisão ficou entre 75% e 95%.

3.2. Questão 2: Quais os indícios de que um aluno pode ser reprovado, baseado em dados gerados pela avaliação formativa?

A possibilidade de reprovação pode ser prognosticada através de variáveis preditivas, e com estes indícios descritos, pode-se dar ao professor a possibilidade de readequar sua metodologia e traçar novas estratégias de ensino para que possa evitar tais reprovações. Em [Liao et al. 2016], com os dados obtidos, foi criado um modelo capaz de classificar os alunos e identificar a situação destes, com uma precisão de 70%, utilizando os dados de testes e treinamentos das 3 primeiras semanas.

O framework PISA, utilizado em [Lishinski et al. 2016], possibilita medir a capacidade do aluno de resolução de problemas, sendo parte essencial do aprendizado de programação. Ele deixa claro que a programação é um conhecimento hierárquico, em que cada conteúdo assimilado serve de base para os posteriores. De forma semelhante, em [Ahadi 2016], foi realizada a coleta dos dados com uma ferramenta capaz de capturar *snapshots* dos códigos-fonte gerados, podendo apresentar ao professor as dificuldades dos alunos. Em [Castro-Wunsch et al. 2017], a partir de dados obtidos e processados por uma RNA, obteve-se um bom resultado na predição do sucesso dos alunos, com resultado de 80-85% de variação em sua eficiência. A ferramenta apresentada em [Estey et al. 2017] tem o diferencial de retornar um *feedback* para o aluno, de forma que, além do professor, o aluno também pode saber como está sua compreensão em relação aos conteúdos.

Algumas variáveis preditoras, exploradas por [Lambert 2015], são: o histórico acadêmico do aluno, fatores matemáticos e experiências anteriores dos alunos, porém ele só encontrou uma correlação forte com o CS0 (*Computer Science 0 – Curso Prévio*) no sucesso do CS1, enquanto não teve sucesso com as demais variáveis utilizadas. No trabalho de [Watson et al. 2014], é feito um teste de 50 diferentes preditores de desempenho, sendo 38 tradicionais e 12 dinâmicos, ambos baseados no comportamento de programação, tendo uma maior eficácia nos cursos de CS1.

Através de um ambiente de virtual, [Silva et al. 2014] mostram que os dados gerados na plataforma Moodle, utilizando o seu algoritmo proposto são excelentes variáveis produtoras tendo uma precisão de 72%, mostrando assim o desempenho da utilização de ambiente virtuais no ensino e a capacidade preditiva da árvore de decisão. Outro traba-

lho que também usa ambiente virtual é o de [Detoni et al. 2014], que a partir dos dados gerados pelo alunos, produziram modelos dos quais foram avaliados pelo WEKA, assim utilizando vários algoritmos de inteligência artificial como Rede Bayesiana, Rede Neural, J48 e Random Forest, apresentados os resultados obtidos dos mesmos tendo uma média de acurácia superior a 90% e com uma precisão entre 75% e 95%. [Brito et al. 2014] utilizam-se de dados educacionais para apresentar uma abordagem que usa algoritmos de aprendizagem acoplados para integrar diversas técnicas de aprendizado e mineração de dados, tais algoritmos como NaiveBayes e J48, mostrando mais um pouco a relevância do algoritmo de inteligência artificial como ferramenta no ensino.

[Gotardo et al. 2013] avaliam a relação entre as notas de ingresso do aluno com seu desempenho ao decorrer do curso, utilizando os algoritmos presentes na ferramenta WEKA tais como Naive Bayes, IBk, SMO, Random Forest e Multi Perceptron, e obteve um resultado superior a 70% com essa ferramenta, podendo se notar a relação entre as notas de ingresso e as notas ao decorrer do primeiro período de curso. [Amorim et al. 2008] mostram as principais fases na implementação de um sistema de previsão e fez o teste de três classificadores amplamente utilizados e mostra as estatísticas referentes a evasão em cada curso, e demonstra a eficácia das técnicas de Machine Learning presentes na ferramenta WEKA. [Ferreira 2016] propõem um modelo para predição de grupos de risco com técnicas de árvore de decisão, e obteve resultados mostrando a possibilidade de interceptação dos dados gerados pelo uso dos métodos de predição, pois métodos, como RNA possuem como deficiência justamente a dificuldade de identificar as causas que levam aos resultados das predições.

3.3. Questão 3: Quais as formas que existem para impedir esta reprovação?

As formas existentes para impedir a reprovação do aluno são unanimemente abordadas da mesma forma nos artigos, pois o objetivo é prever o resultado dos alunos durante alguma disciplina no curso de Ciência da Computação, ao detectar padrões que auxiliem o processo de classificação destes alunos em grupos com desempenhos semelhantes. Estas formas servem para que os professores possam criar uma estratégia para realocar os fatores da disciplina, baseado nos resultados obtidos para disseminar, de forma clara e consistente, todos os fundamentos necessários para formação de um bom profissional. O professor, tendo em mãos a relação dos alunos que possuem possibilidade de reprovação, pode reorganizar a forma que ministra sua aula, oferecer auxílio-extra aos alunos, ou qualquer outra estratégia para evitar estes desempenhos insatisfatórios.

4. Considerações Finais

O mapeamento sistemático teve como foco os métodos para identificação de aluno em situação de risco de reprovação e a correlação entre os resultados obtidos no período inicial da disciplina com a nota final do exame. As questões de pesquisa foram criadas com o intuito de descobrir quais métodos foram utilizados pelos principais autores na área e a eficácia destes.

A questão 1 mostra vários métodos na identificação de alunos com possibilidade de reprovação, concluindo que os mais eficientes para tal são os métodos de estatística descritiva, dentre eles o mais eficaz e mais utilizado é o método de Correlação Produto-Momento de Pearson. A questão 2 analisa os possíveis indícios de reprovação apresentados durante o período do curso, concluindo-se que, a partir da classificação obtida através

dos métodos apresentados para a análise dos dados gerados, poderia se prever o desempenho final dos alunos. A questão 3 analisa as formas de impedir a reprovação, já que através dos métodos avaliados neste artigo, o professor teria em mãos, dados necessários para traçar novas estratégias de ensino.

Esta pesquisa teve por objetivo o levantamento de métodos que prevejam o desempenho final dos alunos, para que uma intervenção seja feita, a fim de evitar eventuais reprovações. Este mapeamento tem como ênfase especial a identificação de métodos que utilizem dados oriundos de metodologias de ensino que adotem avaliação formativa contínua, como a Instrução pelos Colegas. Um ponto importante, que foi possível observar durante este mapeamento, é que métodos estatísticos descritivos alcançam níveis preditivos altos - maior ou igual a 80% - em relação a outros métodos utilizados em trabalhos correlatos.

Referências

- Ahadi, A. (2016). Early identification of novice programmers' challenges in coding using machine learning techniques. In *Proceedings of the 2016 ACM Conference on International Computing Education Research*, pages 263–264. ACM.
- Ahadi, A., Lister, R., and Vihavainen, A. (2016). On the Number of Attempts Students Made on Some Online Programming Exercises During Semester and their Subsequent Performance on Final Exam Questions. *ACM Conference on Innovation and Technology in Computer Science Education*, pages 218–223.
- Amorim, M. J. V., Barone, D., and Mansur, A. U. (2008). Técnicas de Aprendizado de Máquina Aplicadas na Previsão de Evasão Acadêmica. *fBrazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 1(1):666–674.
- Aranha, Á. (2017). Orientação de estágios pedagógicos: avaliação formativa versus avaliação somativa. *Boletim Sociedade Portuguesa de Educação Física*, (7-8):157–165.
- Blando, A. (2015). Dificuldades acadêmicas que interferem na aprendizagem de estudantes universitários de engenharias e de ciências exatas: um estudo fundamentado na epistemologia genética. Dissertação de Mestrado em Educação, UFRGS (Universidade Federal do Rio Grande do Sul), Porto Alegre, Brasil, 126 p.
- Brito, D. M., Almeida Jr, I. A., Queiroga, E. V., and Rêgo, T. G. (2014). Predição de desempenho de alunos do primeiro período baseado nas notas de ingresso utilizando métodos de aprendizagem de máquina. 25(1):882.
- Castro-Wunsch, K., Ahadi, A., and Petersen, A. (2017). Evaluating Neural Networks as a Method for Identifying Students in Need of Assistance. *ACM Technical Symposium on Computer Science Education*, pages 111–116.
- Crouch, C. H. and Mazur, E. (2001). Peer instruction: Ten years of experience and results. *American Journal of Physics*, 69(9):970–977.
- Danielsiek, H. (2016). Stay on These Roads : Potential Factors Indicating Students ' Performance in a CS2 Course. *47th ACM Technical Symposium on Computer Science Education*, pages 12–17.

- Detoni, D., Araujo, R. M., and Cechinel, C. (2014). Predição de reprovação de alunos de educação a distância utilizando contagem de interações. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, 25(1):896.
- Estey, A., Keuning, H., and Coady, Y. (2017). Automatically Classifying Students in Need of Support by Detecting Changes in Programming Behaviour. *ACM Technical Symposium on Computer Science Education*, pages 189–194.
- Ferreira, J. L. C. (2016). Criação e utilização de um MOOC (Massive Open Online Course) como recurso didático no ensino-aprendizagem de algoritmos e programação. Monografia de Mestrado em Ciência da Computação, inos (Universidade do Vale do Rio dos Sinos), São Leopoldo - RS, Brasil, 76 p.
- Gotardo, R., Cereda, P. R. M., and Junior, E. R. H. (2013). Predição do desempenho do aluno usando sistemas de recomendação e acoplamento de classificadores. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, 24(1):657.
- Haden, P., Parsons, D., Wood, K., and Gasson, J. (2017). Student Affect in {CS1}: Insights from an Easy Data Collection Tool. *17th Koli Calling Conference on Computing Education Research*, pages 40–49.
- Holmboe, C., McIver, L., and George, C. (2001). Research agenda for computer science education. In *13th Workshop of the Psychology of Programming Interest Group*, volume 207223.
- INEP (2018). Sinopse estatística do censo da educação superior 2017. Instituto Nacional Estudo e Pesquisas Educacionais Anísio Teixeira. Disponível em <<<http://portal.inep.gov.br/basica-censo-escolar-sinopse-sinopse>>>. Acesso em 22 de novembro de 2018.
- Lambert, L. (2015). Factors That Predict Success in CS1. *J. Comput. Sci. Coll.*, 31(2):165–171.
- Liao, S. N., Zingaro, D., Laurenzano, M. A., Griswold, W. G., and Porter, L. (2016). Lightweight, Early Identification of At-Risk CS1 Students. *ACM Conference on International Computing Education Research*, pages 123–131.
- Lishinski, A., Yadav, A., Enbody, R., and Good, J. (2016). The Influence of Problem Solving Abilities on Students' Performance on Different Assessment Tasks in CS1. *47th ACM Technical Symposium on Computing Science Education - '16*, pages 329–334.
- Müller, M. G., Araujo, I. S., Veit, E. A., and Schell, J. (2017). Uma revisão da literatura acerca da implementação da metodologia interativa de ensino peer instruction (1991 a 2015). *Revista brasileira de ensino de física. São Paulo. Vol. 39, n. 3 (jul./set. 2017)*, e3403, 20 p.
- Porter, L. and Zingaro, D. (2014). Importance of early performance in CS1. *45th ACM technical symposium on Computer science education - '14*, pages 295–300.

- Quille, K. and Bergin, S. (2018). Programming: predicting student success early in CS1. a re-validation and replication study. *23rd Annual ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE 2018*, pages 15–20.
- Ramos, E. d. S. and Brasil, M. M. A. (2012). Um mapeamento sistemático sobre padrões de software para reengenharia de sistemas.
- Robins, A. (2015). The ongoing challenges of computer science education research. *Computer Science Education*, 25(2):115–119.
- Silva, J. M. C., Andrade, F. G., Tessari, R., and Preissler Jr., S. (2014). Alunos em risco: como identificá-los por meio de um ambiente virtual de aprendizagem? *Congresso Brasileiro de Ensino Superior a Distância - ESUD*.
- Watson, C., Li, F. W., and Godwin, J. L. (2014). No Tests Required: Comparing Traditional and Dynamic Predictors of Programming Success. *45th ACM technical symposium on Computer science education*, pages 469–474.