Instituto de Pesquisas Tecnológicas do Estado de São Paulo

Marcelo de Rezende Martins

Uso de redes neurais recorrentes na detecção de padrões de erros cometidos por alunos novatos durante a aprendizagem de programação

São Paulo

2018

Marcelo de Rezende Martins

Uso de redes neurais recorrentes na detecção de padrões de erros cometidos por alunos novatos durante a aprendizagem de programação

Dissertação de Mestrado apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT, como parte dos requisitos para a obtenção do título de Mestre em Engenharia da Computação

Data da aprovação \_\_\_\_/\_\_\_\_\_/\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof.Dr. Marco Aurélio Gerosa (Orientador)

USP – Universidade de São Paulo

Membros da Banca Examinadora:

Prof. Dr. Marco Aurélio Gerosa (Orientador)

USP – Universidade de São Paulo

Prof. Dr. YYYY (Membro)

ZZZZ.

Prof. WWWW (Membro)

ZZZZZZ

Marcelo de Rezende Martins

Uso de redes neurais recorrentes na detecção de padrões de erros cometidos por alunos novatos durante a aprendizagem de programação

Pesquisa apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT, como requisito para a obtenção do título de Mestre em Engenharia da Computação.

Área de Concentração: Engenharia de Software

Orientador: Prof. Dr. Marco Aurélio Gerosa

São Paulo

Abril/2018

Ficha Catalográfica  
Elaborada pelo Centro de Informação Tecnológica do   
Instituto de Pesquisas Tecnológicas do Estado de São Paulo – IPT

# 

# A555p

PEDIR POR EMAIL PARA MARIA DARCI [mdarci@ipt.br](mailto:mdarci@ipt.br) ou telefone 3767 4139.

05-63 CDU 004.41(043)

**Resumo**

Reumo

Palavras-chave: redes neurais recorrentes, aprendizagem de máquina, engenharia de atributos, dificuldades, erros, ciências da computação e educação

Keywords: recurrent neural networks, machine learning, feature engineering, difficulties, errors, ai for education, computer science education.

LISTA DE FIGURAS

[Figura 1 – Visão geral do Processo de revisão sistemática 23](#_Toc513572490)

[Figura 2 - Esquema da construção das palavras-chaves 26](#_Toc513572491)

[Figura 3 - Construção da String de Busca 27](#_Toc513572492)

[Figura 4 - Seleção preliminar dos artigos 30](#_Toc513572493)

LISTA DE QUADROS

No table of figures entries found.

LISTA DE TABELAS

[Tabela 1 - Critérios PICOC da Principal questão 26](#_Toc513663955)

[Tabela 2 - Critérios PICOC da Questão Secundária 26](#_Toc513663956)

[Tabela 3 - Critérios de inclusão e exclusão de estudos 29](#_Toc513663957)

[Tabela 4 - Seleção Preliminar da SCOPUS 47](#_Toc513663958)

[Tabela 5 - Seleção preliminar da Web Of Science 51](#_Toc513663959)

[Tabela 6 - Seleção preliminar da IEEE 53](#_Toc513663960)

[Tabela 7 - Seleção preliminar da ACM 55](#_Toc513663961)

LISTA DE ABREVIATURAS E SIGLAS

MOOC – Curso online, massivo e aberto (Massive Open Online Course)

PICOC – População, Intervenção, Controle, Resultado, Aplicação (Population, Intervention, Control, Output, Context)

SUMÁRIO

[1 Introdução 13](#_Toc513572765)

[1.1 Apresentação e contextualização do problema 16](#_Toc513572766)

[1.2 Objetivo 18](#_Toc513572767)

[1.3 Contribuições Esperadas 19](#_Toc513572768)

[1.4 Método de trabalho 20](#_Toc513572769)

[1.5 Organização do trabalho 20](#_Toc513572770)

[1.7 Cronograma 21](#_Toc513572771)

[2. Conceitos e Trabalhos Relacionados 22](#_Toc513572772)

[2.1 Aprendizagem de Máquina 22](#_Toc513572773)

[2.1.1 Redes Neurais 22](#_Toc513572774)

[2.1.2 Aprendizagem de Máquina Profundo 22](#_Toc513572775)

[2.2 Big Code 23](#_Toc513572776)

[2.3 Trabalhos Relacionados 23](#_Toc513572777)

[2.3.1 Planejamento 24](#_Toc513572778)

[2.3.1.1 Objetivo 25](#_Toc513572779)

[2.3.1.2 Principal Questão 25](#_Toc513572780)

[2.3.1.2.1 Questão Secundária 26](#_Toc513572781)

[2.3.1.3 Palavras-chaves e Sinônimos 27](#_Toc513572782)

[2.3.1.4 Critérios de Seleção de Fontes 28](#_Toc513572783)

[2.3.1.5 Critérios de Seleção dos Estudos 29](#_Toc513572784)

[2.3.1.6 Processo de Seleção 30](#_Toc513572788)

[2.3.2 Condução da Revisão Sistemática 31](#_Toc513572789)

[3 Modelo Para Código-Fonte 32](#_Toc513572790)

[3.1 RNN 32](#_Toc513572791)

[3.2 GRU 32](#_Toc513572792)

[3.3 Modelo 32](#_Toc513572793)

[4 Arquitetura do Sistema para Detecção de Erros 33](#_Toc513572794)

[5 Conclusão 34](#_Toc513572795)

[Apêndice A: Resultados das buscas nas bibliotecas eletrônicas 34](#_Toc513572796)

# Introdução

Em 1936, Alan Turing publicou um artigo chamado *On Computable Numbers, with an application to the Entscheidungsproblem* (TURING, 1936, p. 230—265) no qual descreveu uma máquina teórica capaz de resolver qualquer problema que pudesse ser descrito através de instruções pré-definidas numa fita de papel. Este artigo tornaria-se a base para a computação e invenção dos computadores. Durante a segunda guerra, Turing construiu uma máquina capaz de quebrar a maioria do código Morse secreto usado nas comunicações pela força naval alemã. Segundo alguns especialistas, a guerra teria durado mais tempo se não fosse a máquina inventada por Alan Turing para quebrar a criptografia das mensagens. A humanidade deu um grande salto graças ao avanço dos computadores desde o fim da segunda guerra até os dias de hoje. Este salto tecnológico permitiu o avanço em diversas áreas como saúde, telecomunicações, aviação, engenharia, financeira etc.

Com o aumento do poder de processamento computacional dobrando a cada 18 meses, Lei de Moore, novos avanços em tecnologia surgem como a Internet da Coisas (IoT), cidades inteligentes (Smart Cities), Indústria 4.0. Todas tem um ponto em comum: computação úbiqua, uso de inteligência artificial e big data. E um outro ponto é a automação. Governos em todo o mundo estão preocupados com o advento da inteligência artificial e automação, pois muitos especialistas dizem que haverá um desemprego em massa. A maioria dos empregos que tem uma tarefa repetitiva, que não exige criatividade e interação social, serão automatizados. Em sua maioria, são empregos exercidos por pessoas com baixa qualificação (SMIT, 2016, p. 26). Segundo um relatório da OCDE, este desemprego em massa não ocorrerá tão rapidamente, pois o computador consegue fazer somente tarefas bem definidas e a maioria dos trabalhos não tem uma definição clara das tarefas a serem feitas (ARNTZ. Melanie; GREGORY. Terry; ZIERAHN. Ulrich., 2016, p. 22).

Em um item todos os governos concordam: há uma necessidade em preparar a nova geração para esta nova era. Hoje há um defícit de programadores em todo o mundo. E com o advento do IoT, Indústria 4.0, a demanda por desenvolvedores e programadores irá aumentar ainda mais. Governos de todo o mundo estão empenhados em um projeto de ensino de programação no currículo básico das escolas para atender a esta demanda. Alan Perlis disse em 1960 que programação deveria ser ensinada a todos. O modo de pensar "algoritmicamente" é útil para diversos problemas em nosso cotidiano. E com a popularização e a presença em massa dos computadores em nosso cotidiano, a importância de saber programar aumenta. Governos e empresas multinacionais juntaram forças para ensinar programação à populacao através de iniciativas como o code.org, udacity, scratch etc.

Mas aprender a programar não é uma tarefa fácil. Diversos estudos apontam para a alta taxa de evasão dos alunos dos cursos de Ciência da Computação (WATSON, Christopher; LI, Frederick W.B., 2014, p. 39—44). E outros estudos apontam a alta taxa de reprovação por parte dos alunos nos cursos de introdução à programação (WATSON, Christopher; LI, Frederick W.B., 2014, p. 39—44; BOSSE, Yorah; GEROSA, Marco Aurélio, 2015, p. 1389--1397). Segundo um estudo preliminar feito por Bosse (2015), a taxa de reprovação nos cursos introdutórios de programação ultrapassam a taxa de 30% somente na Universidade de São Paulo. A reprovação no curso de introdução à programação é um dos fatores que contribuem para a evasão do curso. Segundo os dados do censo do Ensino Superior divulgado pelo MEC em 2016, a taxa de evasão média para cursos de bacheralado é em torno de 22%, enquanto para os cursos de bacharelado em Computação, a taxa de evasão supera 28%.

Segundo Kalelioglu (2016), o pensamento computacional é um processo para resolução de problemas, envolve diversos modelos mentais como abstração, reconhecimento de padrões, decomposição, generalização etc. Além da necessidade de desenvolver o pensamento computacional, é importante saber quais são as principais dificuldades encontradas pelos alunos.

Segundo um estudo feito por Brown e Altadmri (2017), a maioria dos instrutores e professores não conseguem identificar quais são os erros mais frequentes cometidos pelos estudantes. A partir dos erros, é possível elaborar materiais e alterar o ritmo da aula para um melhor aproveitamento do aluno.

Com o aumento do poder computacional, crescente uso de ferramentas de aprendizagem de máquina e a grande oferta de repositórios de código-fontes como Github, por exemplo, nasce o termo Big Code (BHATIA, Sahil; SINGH, Rishabh, 2016). Big Code refere-se ao conhecimento adquirido a partir da análise dos repositórios de código fonte através de ferramentas de aprendizagem de máquina. Construir ferramentas que sejam capazes de aprender a partir dos repositórios de código fonte é um novo desafio. Com o conhecimento que podemos adquirir a partir da análise dos repositórios, podemos construir ferramentas para analisar a arquitetura de um projeto, verificar se o código fonte ou o nome do método está de acordo com a convenção, detectar erros de execução de forma antecipada.

Com as ferramentas de aprendizagem de máquina, é possível também identificar os erros mais comuns cometidos pelos alunos. Esta técnica é utilizada para dar uma resposta automática para o aluno, permitindo que ele receba uma avaliação mais rápida que a dada pelo professor ou monitor.

Diversos estudos apontam redes neurais como uma técnica promissora na criação de ferramentas para detectar e corrigir erros de sintaxe, erros de semântica (WANG, 2017) . Um texto publicado no site da Microsoft, mostrou como o uso do algoritmo de *Deep Learning* foi eficaz para identificar as correções necessárias no código fonte submetido pelo aluno.

## Apresentação e contextualização do problema

A oferta de cursos MOOC para ensinar programação aumentou exponencialmente nos últimos 10 anos (adicionar referência). Há diversos fatores que contribuiram para isto. Entre eles a popularização do acesso à internet banda larga e a crescente demanda por programadores ao redor do mundo (adicionar referência). As plataformas MOOC mais populares hoje que ensinam programação são Coursera, EDx, Udacity e Codecademy. Nestas plataformas são disponibilizados aos alunos vídeos, textos e também exercícios para praticar o conteúdo visto. Há outras plataformas que são um repositório de exercícios. O HackerRank e o URIOnlineJudge, por exemplo, contém milhares de exercícios sobre uma variedade enorme de tópicos de computação.

O ensino a distância tem vários desafios e um deles é como reter o aluno. A porcentagem de abandono dos cursos online é alto, quando comparado com os cursos presenciais (verificar se isso é verdade). Entre os motivos da desistência do curso tem a falta de *feedback* e a demora na correção (acrescentar mais motivos e referências). Os exercícios são uma importante ferramenta no aprendizado do aluno. E a correção dos exercícios é uma oportunidade para que o aluno saiba a causa e o motive pelo qual errou.

Segundo XXXX, nos cursos de programação MOOC, a correção e o *feedback* do exercícios são feitos da seguinte maneira:

* Correção automática
* Semi-automática
* Manual

Na correção automática, a nota do aluno é calculada de acordo com a porcentagem de testes que o exercício acertou (é verdade isso?). Neste caso, o *feedback* é providenciado a partir de um mapeamento dos testes e possíveis erros associados (achar referências....). Já na semi-automática, o professor e/ou monitor corrigem os exercícios individualmente. No momento em que o professor provê um *feedback* a um aluno, o sistema já verifica códigos-fontes similares e já envia o mesmo *feedback* para outros alunos (adicionar a referência). Já na manual, os exercícios são corrigidos e *feedbacks* são providenciados individualmente pelo professor e/ou monitor.

Neste contexto, surgiram soluções de correção automática e semi-automática para plataformas MOOC utilizando inteligência artificial. Segundo um estudo XXXX, as principais soluções de correções de exercícios de programação automático são baseados em clustering, utilizando o algoritmo KNN, por exemplo (adicionar mais exemplos, referências). Recentemente, surgiram soluções utilizando aprendizagem de máquina profundo, como o sk\_p, DeepFix, SynFix entre outros para identificar erros de programação (WANG, 2017). Tanto o SynFix quanto o DeepFix utlizam redes recorrentes. E ambos utilizam sequência de *tokens* como forma de representação da entrada de dados para treinar a rede neural (refazer este trecho usando o artigo do WANG).

Segundo Wang (2017), as redes recorrentes mostraram-se promissoras na identificação e correção de erros em códigos-fontes. Porém, a representação dos programas através de uma sequência de *tokens* dificulta a aprendizagem pela rede neural. Segundo Wang (2017), a diferença na sintaxe de um programa correto e errado é de apenas poucos caracteres, na maioria dos casos. Esta diferença mínima, dificulta a rede neural construir o melhor modelo para diferenciar os programas corretos e programas errados.

Wang (2017) propôs uma forma de representar o

Esta dissertação tentará responder a seguinte pergunta:

1. Como as redes neurais recorrentes podem ser utilizadas para identificar padrões de erros de semântica cometidos por alunos?

## Objetivo

O uso de algoritmo de aprendizagem para correção automática de exercícios não é novidade. Porém, boa parte dos estudos foram feitos para plataformas MOOC, no qual a maioria dos exercícios são avaliados por meio de uma base de testes e a nota representa a porcentagem de testes que o exercício passou. Neste estudo, identificaremos os padrões de erros por meio do uso do algoritmo de redes neurais aplicado num conjunto de exercícios previamente avaliados por monitores e professores. A avaliação feita por monitores e instrutores leva em consideração se o aluno entendeu conceitos como instruções de repetição, instruções de decisão, chamada de métodos, recursão, uso de vetores, ponteiros, passagem de parâmetros etc. Este tipo de avaliação é diferente das avaliações baseadas em porcentagem de testes.

O objetivo da pesquisa é verificar o uso das redes neurais recorrentes como técnica para detecção de padrões de erros cometidos por alunos previamente analisados por monitores e instrutores. O escopo deste estudo será limitado inicialmente a uma classe de problema. Normalmente, os exercícios feitos pelos alunos visam reforçar um conceito visto em sala de aula. Por exemplo, o propósito de um programa para verificar se um ano é bissexto é reforçar os conceitos de instrução de decisão, operadores booleanos. Já um programa para exibir um calendário reforça os conceitos de instrução de decisão, instrução de repetição, operadores booleanos, formatação da saída, porém não exige o conhecimento de matrizes, por exemplo. Neste estudo, as redes neurais recorrentes serão utilizadas para detectar padrões de erros de lógica em códigos-fontes feitos em Python, inicialmente.

## Contribuições Esperadas

Conforme mencionado anteriormente, o contexto do uso de redes neurais recorrentes na detecção de padrões de erros será a principal contribuição deste trabalho. Contribuições secundárias serão o levantamento de padrões de erros de lógica cometidos por alunos novatos durante a aprendizagem de programação, definição de classe de problemas e exercícios, definição dos atributos que serão levados em consideração para o treinamento, análise e validação do uso de redes neurais recorrentes para detecção de padrões de erros no contexto deste trabalho.

## Método de trabalho

## Organização do trabalho

## Cronograma

# Conceitos e Trabalhos Relacionados

## Aprendizagem de Máquina

## Redes Neurais

## Aprendizagem de Máquina Profundo

## Big Code

## Trabalhos Relacionados

Para identificar os trabalhos relacionados, foi conduzido uma revisão sistemática. Segundo Bapuji et al. (2007 apud WALKER, 2010), a revisão sistemática aumenta a qualidade do processo de revisão estabelecendo uma revisão sistemática, transparente e reproduzível da literatura.

Segundo Walker (2010), uma revisão sistemática deve incluir objetivos claros, ser reproduzível, ter uma busca ampla e inclusiva, quer dizer, a seleção deve ser baseada no mérito do trabalho, a fim de reduzir o viés cognitivo. Além disso, deve incorporar um método de síntese para organizar a literatura.

Para apoiar a revisão sistemática foi utilizado a ferramenta StArt desenvolvido pelo Laboratório de Pesquisa em Engenharia de Software da Universidade Federal de São Carlos (UFSCar). De acordo com o guia de revisão sistemática para engenharia de software desenvolvido por Kitchenham (2007), a revisão sistemática é composta por três fases: Planejamento, Condução e Documentação da revisão.

Na fase de planejamento, é definido o objetivo da revisão e é criado o protocolo contendo as informações importantes da revisão sistemática. Na fase de condução, é identificada e feita a seleção dos estudos primários relevantes, extração e síntese dos dados. E, por fim, na fase de documentação, é feito a descrição, divulgação e avaliação dos resultados relatados (KITCHENHAM; CHARTERS, 2007).

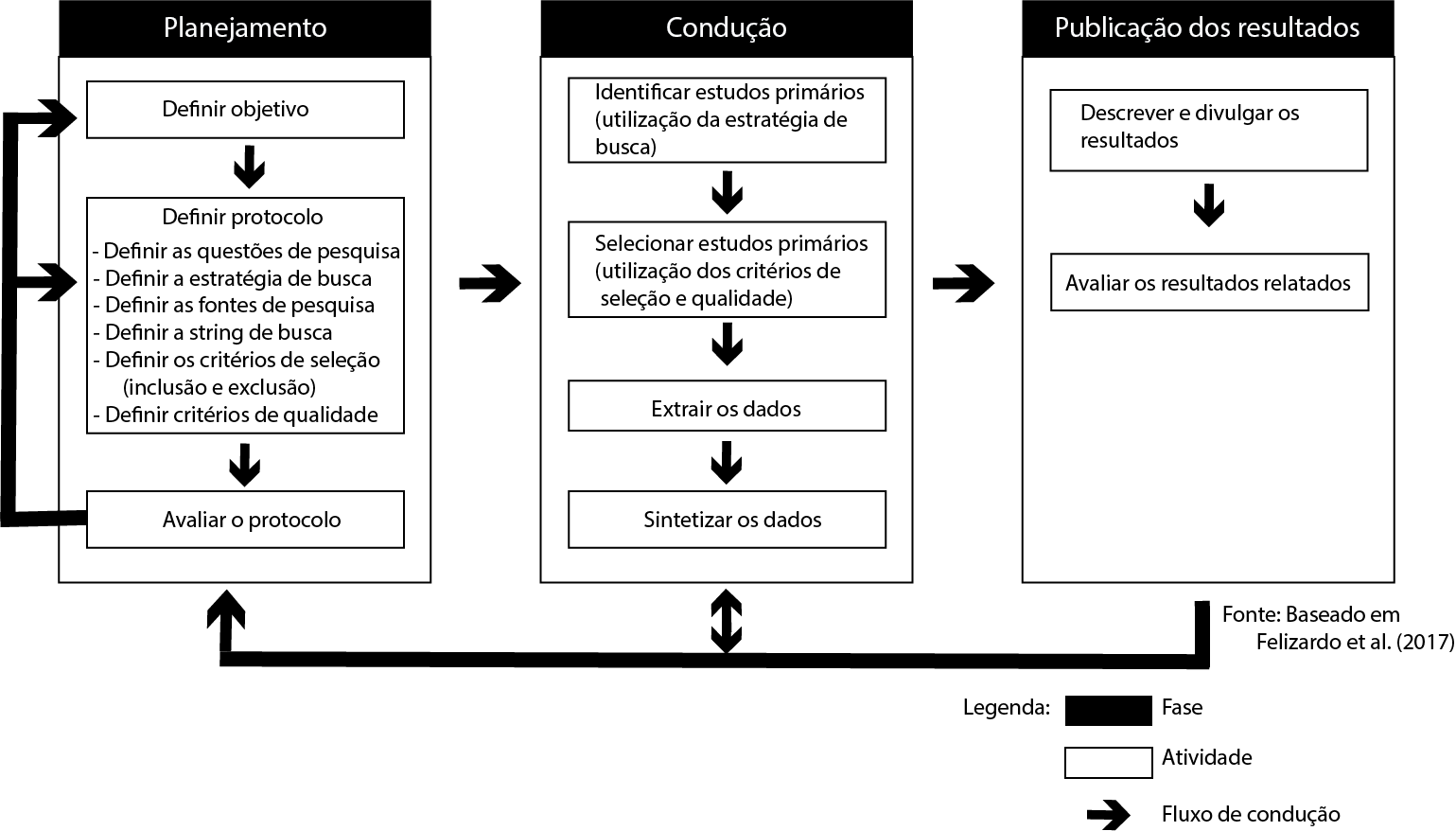


Figura 1 – Visão geral do Processo de revisão sistemática

­

## Planejamento

Segundo Kitchenham e Charters (2007), antes de iniciar a revisão sistemática propriamente, é necessário confirmar qual a necessidade de fazer a revisão. Para isto, é importante definir o objetivo, as questões de pesquisa e um protocolo básico contendo os procedimentos que serão seguidos durante o processo de revisão sistemática.

A ferramenta StArt providencia um protocolo de revisão sistemática baseada no guia desenvolvido por Kitchenham e Charters e um outro guia criado por Biolchini et al. (FABRI, 2016). O planejamento da revisão sistemática foi feito com base neste protocolo providenciado pelo StArt.

## Objetivo

O objetivo principal desta revisão sistemática é avaliar o uso de redes neurais na detecção e correção de erros em códigos-fontes. Especificamente, há um interesse na detecção e correção de erros em códigos-fontes feitos por alunos durante a aprendizagem de programação.

Um outro objetivo derivado é avaliar como representar o código-fonte para que as redes neurais sejam capazes de criar um modelo capaz de identificar os erros de semântica.

## Principal Questão

A principal questão que esta revisão sistemática pretende responder é:

Como as redes neurais recorrentes podem ser utilizadas para detectar erros de semântica em códigos-fontes desenvolvidos por alunos?

Os critérios PICOC foram criados para auxiliar na seleção e exclusão de estudos durante a condução da revisão sistemática:

|  |  |
| --- | --- |
| Critérios PICOC | |
| População | Estudos que usam redes neurais na detecção e correção de códigos fontes; |
| Intervenção | Uso de redes neurais na detecção e correção de códigos fontes; |
| Controle | Não se aplica. |
| Resultado | Estudos primários e secundários sobre o uso de redes neurais na correção e/ou feedback automático de exercícios de programação. |
| Aplicação | Esta revisão contribuirá para ampliar o conhecimento sobre o uso de redes neurais, redes recorrentes e deep learning na detecção e/ou correção de códigos fontes. |

Tabela 1 - Critérios PICOC da Principal questão

## Questão Secundária

Uma pergunta secundária que esta revisão sistemática pretende responder também é:

Como representar os códigos fontes para que a rede neural seja capaz de construir um modelo capaz de identificar e classificar erros de semântica em códigos fontes?

Os critérios PICOC também foram criados para auxiliar na seleção e exclusão de estudos:

|  |  |
| --- | --- |
| Critérios PICOC | |
| População | Estudos que utilizam redes neurais para análise de códigos fontes; |
| Intervenção | Representação de códigos fontes em modelos de redes neurais; |
| Controle | Não se aplica. |
| Resultado | Estudos primários e secundários sobre representação de códigos fontes em modelos de redes neurais. |
| Aplicação | Esta revisão contribuirá para ampliar o conhecimento sobre o como representar os códigos fontes para construir um modelo de rede neural. |

Tabela 2 - Critérios PICOC da Questão Secundária

## Palavras-chaves e Sinônimos

Os estudos de Bhatia e Singh (2016); Wang, Singh e Su (2017); Santos et al. (2017); Denero et al. (2017); Pu et al. (2016); Devlin et al. (2017) serviram como referências para construir as palavras-chaves da string de busca. As figuras 2 e 3 contém um esquema para melhor visualizar como foi construído a string de busca a partir das palavras-chaves.

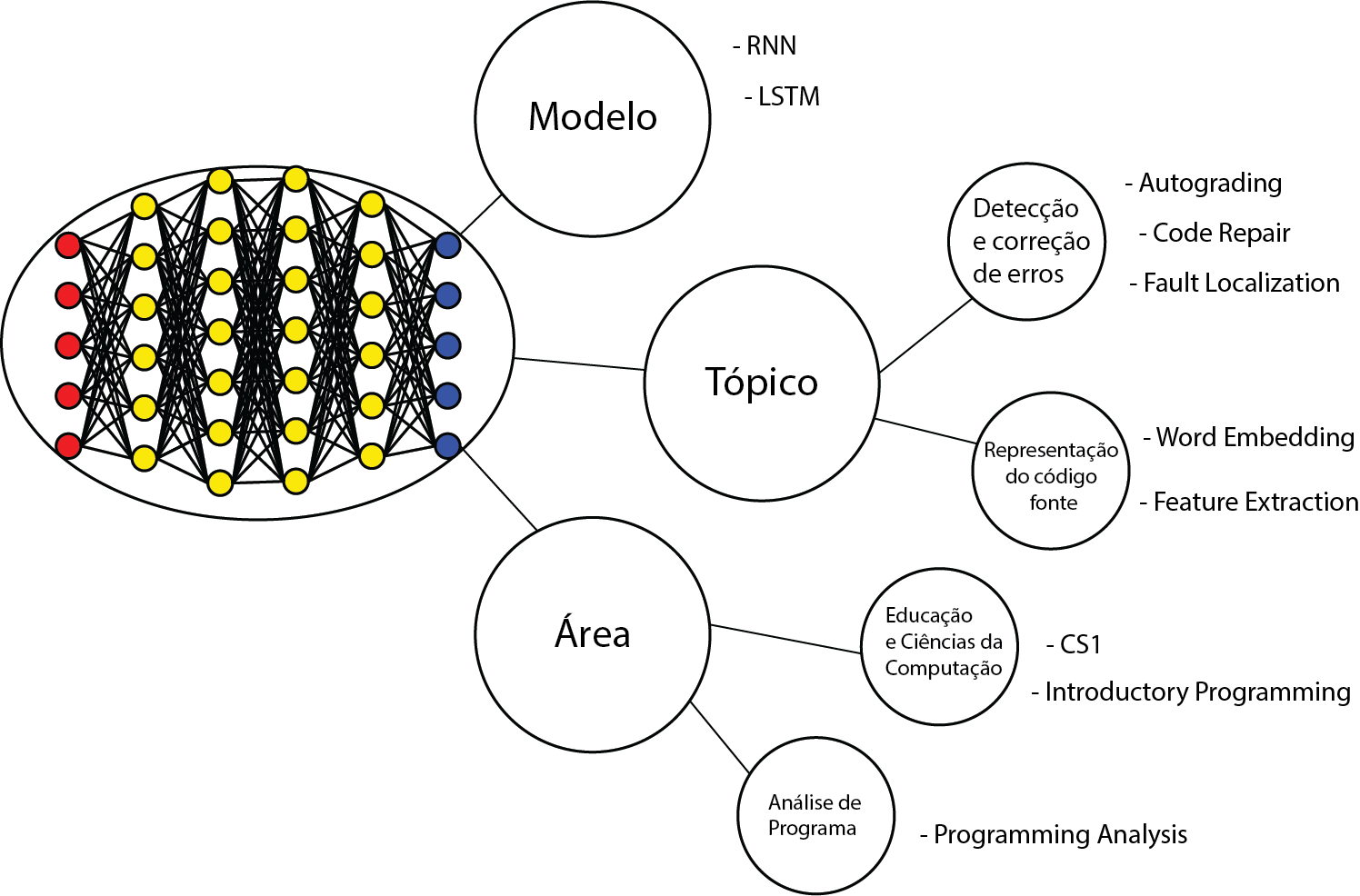


Figura 2 - Esquema da construção das palavras-chaves

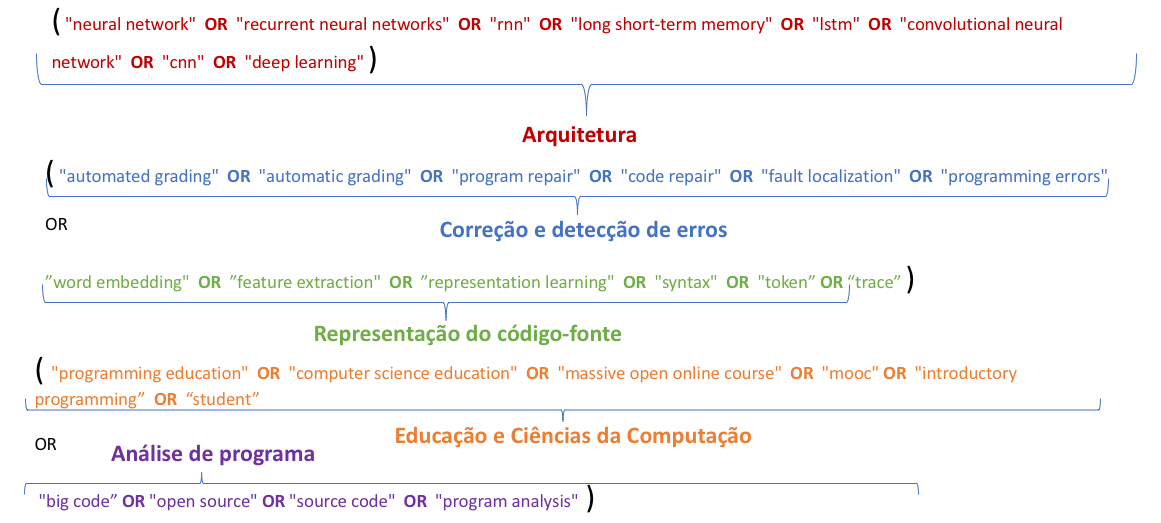


Figura 3 - Construção da String de Busca

## Critérios de Seleção de Fontes

Os critérios de seleção das fontes de pesquisa são (STEINMACHER ET AL., 2015):

* Bibliotecas digitais disponíveis via web que indexem artigos em inglês da área de inteligência articial e/ou educação em ciências da computação;
* Tenha suporte a buscas utilizando expressões booleanas;
* Conceda acesso ao texto completo ou tenha o endereço para acesso ao texto completo.

De acordo com os critérios acima e seguindo a sugestão de Kitchenham e Brereton (2013), as seguintes fontes foram selecionadas:

* Scopus
* Web of Science
* IEEE
* ACM

## Critérios de Seleção dos Estudos

De acordo com Kitchenham e Charters (2007), os critérios de seleção dos estudos servem para identificar os estudos primários que tem uma evidência direta sobre as questões da revisão sistemática. A partir do estudo de Nunes (2017) e da questão de pesquisa, os seguintes critérios de inclusão e exclusão foram definidos na tabela abaixo:

|  |  |
| --- | --- |
| Critérios de Seleção de Estudos | |
| Inclusão | **Exclusão** |
| I1 - Os documentos devem estar disponíveis na web; | E1 - Estudos que não utilizam redes neurais; |
| I2 - Publicações feitas a partir de 2012; | E2 - Trabalhos que não utilizam o código fonte como entrada de dados para o treino supervisionado; |
| I3 - Publicações em Inglês ou Português; | E3 - Publicações anteriores a 2012 que não tratam do modelo de rede neural proposto por esta pesquisa (RNN); |
| I4 - Estudos que utilizam redes neurais para análise do código fonte; | E4 - Pôsteres, tutoriais, relatórios técnicos. Textos que não sejam artigos (curto ou longo), dissertação de mestrado, tese de doutorado, livros, anais de eventos; |

Tabela 3 - Critérios de inclusão e exclusão de estudos

Para um artigo ser aceito, ele deve satisfazer todos os critérios de inclusão. E não deve atender os critérios de exclusão. Além disso, foram excluídos os artigos duplicados.

## Processo de Seleção

Para o processo de seleção dos estudos, foram definidas as mesmas etapas descritas no estudo feito por Steinmacher et al. (2015):

1. Busca nas fontes selecionadas: A busca é feita nas bibliotecas eletrônicas selecionadas. Os artigos e referências são coletados e armazenados para análise.
2. Análise do título, resumo e palavras-chaves: O título, resumo e palavras-chaves são lidos para verificar quais estudos atendem aos critérios de inclusão e exclusão.
3. *Author Snowballing*: Estudos encontrados e selecionados pela busca nas bibliotecas eletrônicas são analisadas em relação aos autores. É feita uma busca por outros artigos publicados pelos mesmos autores. Os artigos encontrados são submetidos a mesma análise de título, resumo e palavras-chaves (passo 2).
4. Análise da introdução e conclusão: A primeira e última seção dos estudos são avaliados em relação aos seus objetivos e resultados. Esta análise permite verificar se os artigos responderam a pergunta de pesquisa e se atenderam a todos os critérios de inclusão e exclusão. Quando a leitura da primeira e última seção são inconclusivas, o artigo inteiro é lido para decidir a sua inclusão ou exclusão.
5. *Backward snowballing sampling*: Estudos encontrados na busca nas fontes selecionadas e pelo *author snowballing* são analisados em relação a suas referências. As referências são avaliadas a partir do mesmo processo dos artigos encontrados nas bibliotecas digitais e encontrados através do *author snowballing*: São analisados os títulos, resumos e palavras-chaves, seguida da análise da introdução e conclusão. É feito apenas 1 nível do *snowballing* para referências. Quer dizer, não são levados em consideração as referências dos artigos que foram encontrados através do *backward snowballing*.
6. Leitura completa: Finalmente, após a análise da introdução e conclusão, é feito uma leitura completa do artigo selecionado para decidir a sua inclusão ou exclusão.

Colocar uma figura

## Condução da Revisão Sistemática

Os documentos foram coletados a partir das buscas feitas nas bibliotecas eletrônicas com as palavras-chaves definidas na seção 2.3.1.3 [Palavras-chaves e Sinônimos](#Palavras-chaves e Sinônimos). As primeiras buscas foram feitas no dia 21 de abril de 2018. E as últimas buscas foram feitas no dia 07 de maio de 2018.

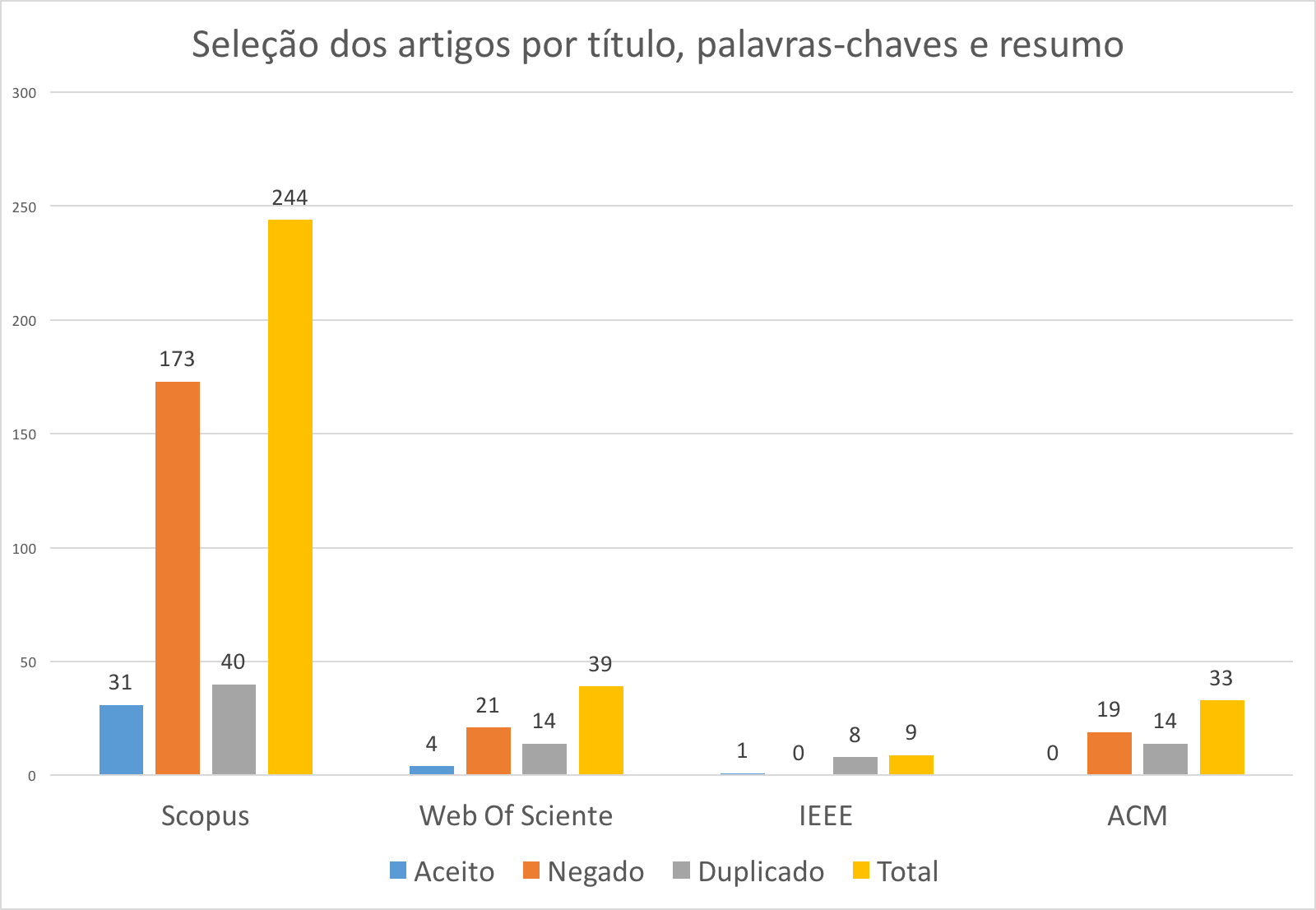


Figura 4 - Seleção preliminar dos artigos

A figura 4 mostra o resultado da seleção preliminar dos artigos. Os artigos foram inicialmente selecionados com base no título, palavras-chaves e resumo. As análises dos artigos seguiram a seguinte ordem: Scopus, Web Of Science, IEEE, ACM. Scopus contribuiu com 31 artigos nesta seleção preliminar, Web Of Science 4, IEEE 1 e ACM não teve artigo selecionado.

O fato da Scopus ter sido a primeira a ser analisada e o fato dela indexar os principais jornais e conferências da IEEE e ACM explica a quantidade de artigos selecionados. Porém, a Scopus tem um atraso na indexação dos artigos em relação a IEEE e ACM. Neste caso, foi possível encontrar um artigo de uma conferência da IEEE que não havia sido indexada ainda pela Scopus.

Outro ponto interessante, é a diferença na quantidade de artigos entre a Scopus e a Web Of Science. Ambas são indexadoras de artigos com ampla cobertura de jornais, revistas e conferências. Analisando os resultados, notou-se que a Scopus contém mais jornais, revistas e conferências na área de aprendizagem de máquina. E, muitas delas, são áreas especializadas da inteligência artificial, como diagnósticos médicos, processamento de imagens, reconhecimento de voz, por exemplo.

O [Apêndice A: Resultados das buscas nas bibliotecas eletrônicas](#Apêndice A: Resultados das buscas nas bibliotecas eletrônicas) contém os resultados das buscas e a avaliação de cada artigo segundo os critérios definidos na seção 2.3.1.5 [Critérios de Seleção de Estudos](#Critérios de Seleção de Estudos).

## Extração Dos Dados (Modificar após ler todos os artigos)

## Síntese

# Modelo Para Código-Fonte

# RNN

# GRU

# Modelo

# Arquitetura do Sistema para Detecção de Erros

# Conclusão

# Apêndice A: Resultados das buscas nas bibliotecas eletrônicas

Os artigos foram coletados entre o dia 21 de abril e 07 de maio de 2018. As buscas foram feitas nas bases da Scopus, Web Of Science, IEEE e ACM. Optou-se por não incluir o endereço do artigo (url) nas tabelas de relação de resultados, pois as bases Web of Science, IEEE e ACM não disponibilizaram tal informação para uma boa parte dos artigos (refazer a busca para verificar este problema).

As tabelas abaixo contém os resultados de busca e a avaliação da fase de seleção preliminar. Nesta fase, foram analisados os títulos, palavras-chaves e resumos dos artigos. A coluna *status/selection* contém o resultado da avaliação do artigo, se ele foi aceito ou rejeitado. Este resultado obedece aos critérios e regras definidas na seção 2.3.1.5 [Critérios de Seleção de Estudos](#Critérios de Seleção de Estudos).

Todos os resultados, arquivos bibtex, arquivos .start (arquivos da ferramenta de apoio a revisão sistemática) estão disponíveis no endereço (mudar o endereço depois): <https://github.com/mrezende/mpq-discipline>

## Scopus

**Data de busca**: 21 de abril, 25 de abril, 07 de maio de 2018

**String de busca**: TITLE-ABS-KEY(( "neural network" OR "recurrent neural networks" OR "rnn" OR "long short-term memory" OR "lstm" OR "convolutional neural network" OR "cnn" OR "deep learning" ) AND ( "automated grading" OR "automatic grading" OR "program repair" OR "code repair" OR "corrector" OR "fault localization" OR "programming errors" OR "word embedding" OR "feature extraction" OR "representation learning" OR "syntax" OR "token" OR "trace" ) AND ( "programming education" OR "computer science education" OR "massive open online course" OR "mooc" OR "introductory programming" OR "student" OR "big code" OR "open source" OR "source code" OR "program analysis" ))

**Campos pesquisados**: Título, Palavras-chaves e Resumo

**Período considerado**: 2012 a maio de 2018

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID Paper | Title | Authors | Year | Status/Selection |
| S1 | Cross-Project Transfer Representation Learning for Vulnerable Function Discovery | Lin, G. and Zhang, J. and Luo, W. and Pan, L. and Xiang, Y. and De Vel, O. and Montague, P. | 2018 | Aceito (I1, I2, I3, I4) |
| S2 | Automatically classifying source code using tree-based approaches | Phan, A.V. and Chau, P.N. and Nguyen, M.L. and Bui, L.T. | 2018 | Aceito (I1, I2, I3, I4) |
| S3 | Machine translation-based bug localization technique for bridging lexical gap | Xiao, Y. and Keung, J. and Bennin, K.E. and Mi, Q. | 2018 | Aceito (I1, I2, I3, I4) |
| S4 | CCLearner: A deep learning-based clone detection approach | Li, L. and Feng, H. and Zhuang, W. and Meng, N. and Ryder, B. | 2017 | Aceito (I1, I2, I3, I4) |
| S5 | Poster: Vulnerability discovery with function representation learning from unlabeled projects | Lin, G. and Zhang, J. and Luo, W. and Pan, L. and Xiang, Y. | 2017 | Aceito (I1, I2, I3, I4) |
| S6 | Source code classification using Neural Networks | Gilda, S. | 2017 | Aceito (I1, I2, I3, I4) |
| S7 | Applying deep learning based automatic bug triager to industrial projects | Lee, S.-R. and Heo, M.-J. and Lee, C.-G. and Kim, M. and Jeong, G. | 2017 | Aceito (I1, I2, I3, I4) |
| S8 | Software defect prediction via convolutional neural network | Li, J. and He, P. and Zhu, J. and Lyu, M.R. | 2017 | Aceito (I1, I2, I3, I4) |
| S9 | Replicating Parser Behavior Using Neural Machine Translation | Alexandru, C.V. and Panichella, S. and Gall, H.C. | 2017 | Aceito (I1, I2, I3, I4) |
| S10 | A memory-Augmented neural model for automated grading | Zhao, S. and Zhang, Y. and Xiong, X. and Botelho, A. and Heffernan, N. | 2017 | Aceito (I1, I2, I3, I4) |
| S11 | SibStCNN and TBCNN + kNN-TED: New models over tree structures for source code classification | Phan, A.V. and Nguyen, M.L. and Bui, L.T. | 2017 | Aceito (I1, I2, I3, I4) |
| S12 | DeepFix: Fixing Common C language errors by deep learning | Gupta, R. and Pal, S. and Kanade, A. and Shevade, S. | 2017 | Aceito (I1, I2, I3, I4) |
| S13 | Source code authorship attribution using long short-term memory based networks | Alsulami, B. and Dauber, E. and Harang, R. and Mancoridis, S. and Greenstadt, R. | 2017 | Aceito (I1, I2, I3, I4) |
| S14 | A Comparison among ARIMA, BP-NN, and Moga-NN for Software Clone Evolution Prediction | Pati, J. and Kumar, B. and Manjhi, D. and Shukla, K.K. | 2017 | Aceito (I1, I2, I3, I4) |
| S15 | Sk-P: A neural program corrector for MOOCs | Pu, Y. and Narasimhan, K. and Solar-Lezama, A. and Barzilay, R. | 2016 | Aceito (I1, I2, I3, I4) |
| S16 | Deep learning code fragments for code clone detection | White, M. and Tufano, M. and Vendome, C. and Poshyvanyk, D. | 2016 | Aceito (I1, I2, I3, I4) |
| S17 | Automatically learning semantic features for defect prediction | Wang, S. and Liu, T. and Tan, L. | 2016 | Aceito (I1, I2, I3, I4) |
| S18 | Convolutional neural networks over tree structures for programming language processing | Mou, L. and Li, G. and Zhang, L. and Wang, T. and Jin, Z. | 2016 | Aceito (I1, I2, I3, I4) |
| S19 | A convolutional attention network for extreme summarization of source code | Allamanis, M. and Peng, H. and Sutton, C. | 2016 | Aceito (I1, I2, I3, I4) |
| S20 | Learning program embeddings to propagate feedback on student code | Piech, C. and Huang, J. and Nguyen, A. and Phulsuksombati, M. and Sahami, M. and Guibas, L. | 2015 | Aceito (I1, I2, I3, I4) |
| S23 | Electronic proofs in mathematics education - A South African Teacher Professional Development (TPD) course informing the conceptualisation of an e-proof system authoring support workshop | Platz, M. and Krieger, M. and Niehaus, E. and Winter, K. | 2017 | Rejeitado (E1) |
| S24 | Long short-term memory RNN for biomedical named entity recognition | Lyu, C. and Chen, B. and Ren, Y. and Ji, D. | 2017 | Rejeitado (E2) |
| S25 | Semantically Enhanced Software Traceability Using Deep Learning Techniques | Guo, J. and Cheng, J. and Cleland-Huang, J. | 2017 | Rejeitado (E2) |
| S26 | A hybrid scheme for automated essay grading based on LVQ and NLP techniques | Shehab, A. and Elhoseny, M. and Hassanien, A.E. | 2017 | Rejeitado (E1) |
| S27 | Character-Level Convolutional Networks for Arithmetic Operator Character Recognition | Liang, Z. and Li, Q. and Liao, S. | 2017 | Rejeitado (E2) |
| S28 | AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP | Soliman, A.B. and Eissa, K. and El-Beltagy, S.R. | 2017 | Rejeitado (E2) |
| S30 | Making sense of neural machine translation | Forcada, M.L. | 2017 | Rejeitado (E2) |
| S33 | Bot.zen @ EVALITA 2016 - A minimally-deep learning PoS-tagger (trained for Italian Tweets) | Stemle, E.W. | 2016 | Rejeitado (E2) |
| S40 | Implicit Heterogeneous Features Embedding in Deep Knowledge Tracing | Yang, H. and Cheung, L.P. | 2018 | Aceito (I1, I2, I3, I4) |
| S41 | Defect Prediction in Android Binary Executables Using Deep Neural Network | Dong, F. and Wang, J. and Li, Q. and Xu, G. and Zhang, S. | 2017 | Aceito (I1, I2, I3, I4) |
| S42 | Automatic categorization with deep neural network for open-source Java projects | Nguyen, A.T. and Nguyen, T.N. | 2017 | Aceito (I1, I2, I3, I4) |
| S43 | Bug Localization with Combination of Deep Learning and Information Retrieval | Lam, A.N. and Nguyen, A.T. and Nguyen, H.A. and Nguyen, T.N. | 2017 | Aceito (I1, I2, I3, I4) |
| S44 | Predicting vulnerable software components through deep neural network | Pang, Y. and Xue, X. and Wang, H. | 2017 | Aceito (I1, I2, I3, I4) |
| S45 | Deep knowledge tracing on programming exercises | Wang, L. and Sy, A. and Liu, L. and Piech, C. | 2017 | Aceito (I1, I2, I3, I4) |
| S46 | Deep multitask learning for semantic dependency parsing | Peng, H. and Thomson, S. and Smith, N.A. | 2017 | Aceito (I1, I2, I3, I4) |
| S47 | Feature analysis for duplicate detection in programming QA communities | Zhang, W.E. and Sheng, Q.Z. and Shu, Y. and Nguyen, V.K. | 2017 | Aceito (I1, I2, I3, I4) |
| S48 | VDNS: An algorithm for cross-platform vulnerability searching in binary firmware | Chang, Q. and Liu, Z. and Wang, M. and Chen, Y. and Shi, Z. and Sun, L. | 2016 | Aceito (I1, I2, I3, I4) |
| S49 | Toward deep learning software repositories | White, M. and Vendome, C. and Linares-VÃ¡squez, M. and Poshyvanyk, D. | 2015 | Aceito (I1, I2, I3, I4) |
| S50 | Comparing techniques for authorship attribution of source code | Burrows, S. and Uitdenbogerd, A.L. and Turpin, A. | 2014 | Aceito (I1, I2, I3, I4) |
| S51 | NiftyNet: a deep-learning platform for medical imaging | Gibson, E. and Li, W. and Sudre, C. and Fidon, L. and Shakir, D.I. and Wang, G. and Eaton-Rosen, Z. and Gray, R. and Doel, T. and Hu, Y. and Whyntie, T. and Nachev, P. and Modat, M. and Barratt, D.C. and Ourselin, S. and Cardoso, M.J. and Vercauteren, T. | 2018 | Rejeitado (E2) |
| S52 | A tutorial review on entropy-based handcrafted feature extraction for information fusion | Guido, R.C. | 2018 | Rejeitado (E2) |
| S53 | FMLLR Speaker Normalization with i-Vector: In Pseudo-FMLLR and Distillation Framework | Joy, N.M. and Kothinti, S.R. and Umesh, S. | 2018 | Rejeitado (E2) |
| S54 | Development of an Autonomous Vehicle Control Strategy Using a Single Camera and Deep Neural Networks | Navarro, A. and Joerdening, J. and Khalil, R. and Brown, A. and Asher, Z. | 2018 | Rejeitado (E1) |
| S55 | Deep convolutional neural network for drowsy student state detection | Zhao, G. and Liu, S. and Wang, Q. and Hu, T. and Chen, Y. and Lin, L. and Zhao, D. | 2018 | Rejeitado (E2) |
| S56 | Mammogram classification schemes by using convolutional neural networks | Soriano, D. and Aguilar, C. and Ramirez-Morales, I. and Tusa, E. and Rivas, W. and Pinta, M. | 2018 | Rejeitado (E2) |
| S58 | An overview of automatic speaker verification system | Naika, R. | 2018 | Rejeitado (E2) |
| S60 | An Emotion Recognition Model Based on Facial Recognition in Virtual Learning Environment | Yang, D. and Alsadoon, A. and Prasad, P.W.C. and Singh, A.K. and Elchouemi, A. | 2018 | Rejeitado (E2) |
| S64 | ADAGE signature analysis: Differential expression analysis with data-defined gene sets | Tan, J. and Huyck, M. and Hu, D. and Zelaya, R.A. and Hogan, D.A. and Greene, C.S. | 2017 | Rejeitado (E2) |
| S65 | A facial recognition application based on incremental supervised learning | BuhuÅŸ, E.R. and Grama, L. and Åžerbu, C. | 2017 | Rejeitado (E2) |
| S66 | How deep learning extracts and learns leaf features for plant classification | Lee, S.H. and Chan, C.S. and Mayo, S.J. and Remagnino, P. | 2017 | Rejeitado (E2) |
| S68 | Crafting GBD-Net for Object Detection | Zeng, X. and Ouyang, W. and Yan, J. and Li, H. and Xiao, T. and Wang, K. and Liu, Y. and Zhou, Y. and Yang, B. and Wang, Z. and Zhou, H. and Wang, X. | 2017 | Rejeitado (E2) |
| S69 | Unsupervised t-Distributed Video Hashing and its Deep Hashing Extension | Hao, Y. and Mu, T. and Goulermas, J.Y. and Jiang, J. and Hong, R. and Wang, M. | 2017 | Rejeitado (E2) |
| S70 | NRC: Non-coding RNA Classifier based on structural features | Fiannaca, A. and La Rosa, M. and La Paglia, L. and Rizzo, R. and Urso, A. | 2017 | Rejeitado (E2) |
| S71 | Bilinear Convolutional Neural Networks for Fine-grained Visual Recognition | Lin, T. and RoyChowdhury, A. and Maji, S. | 2017 | Rejeitado (E2) |
| S72 | Improvements in IITG Assamese Spoken Query System: Background Noise Suppression and Alternate Acoustic Modeling | Shahnawazuddin, S. and Thotappa, D. and Dey, A. and Imani, S. and M. Prasanna, S.R. and Sinha, R. | 2017 | Rejeitado (E2) |
| S73 | Modular array-based GPU computing in a dynamically-typed language | Springer, M. and Wauligmann, P. and Masuhara, H. | 2017 | Rejeitado (E1) |
| S74 | Facial expression recognition using general regression neural network | Talele, K. and Shirsat, A. and Uplenchwar, T. and Tuckley, K. | 2017 | Rejeitado (E2) |
| S75 | Approach of detecting low-rate DoS attack based on combined features | Wu, Z.-J. and Zhang, J.-A. and Yue, M. and Zhang, C.-F. | 2017 | Rejeitado (E2) |
| S76 | Bad teacher or unruly student: Can deep learning say something in Image Forensics analysis? | Rota, P. and Sangineto, E. and Conotter, V. and Pramerdorfer, C. | 2017 | Rejeitado (E2) |
| S77 | A Conceptual Model for an Intelligent Simulation-Based Learning Management System Using a Data Mining Agent in Clinical Skills Education | Samra, H.E. and Li, A.S. and Soh, B. and Alzain, M.A. | 2017 | Rejeitado (E2) |
| S78 | Detection of breast cancer from histopathology image and classifying benign and malignant state using fuzzy logic | Johra, F.-T. and Shuvo, M.M.H. | 2017 | Rejeitado (E2) |
| S79 | Heterogeneous Features Integration in Deep Knowledge Tracing | Cheung, L.P. and Yang, H. | 2017 | Rejeitado (E2) |
| S81 | Speech emotion recognition: Recurrent neural networks compared to SVM and linear regression | Kerkeni, L. and Serrestou, Y. and Mbarki, M. and Mahjoub, M.A. and Raoof, K. and Cleder, C. | 2017 | Rejeitado (E2) |
| S82 | Deep neural networks regularization for structured output prediction | Belharbi, S. and HÃ©rault, R. and Chatelain, C. and Adam, S. | 2017 | Rejeitado (E2) |
| S83 | Tracking objects using Artificial Neural Networks and wireless connection for robotics | OÃ±ate, J.M.B. and Chipantasi, D.J.M. and Erazo, N.D.R.V. | 2017 | Rejeitado (E2) |
| S85 | From homogeneous network to neural nets with fractional derivative mechanism | Gomolka, Z. and Dudek-Dyduch, E. and Kondratenko, Y.P. | 2017 | Rejeitado (E2) |
| S86 | Direct modelling of magnitude and phase spectra for statistical parametric speech synthesis | Espic, F. and Valentini-Botinhao, C. and King, S. | 2017 | Rejeitado (E2) |
| S88 | Knowledge-based system in an affective and intelligent tutoring system | Cabada, R.Z. and Estrada, M.L.B. and PÃ©rez, Y.H. | 2017 | Rejeitado (E2) |
| S91 | Handwriting: Recognition, development and analysis | Bezerra, B.L.D. and Zanchettin, C. and Toselli, A.H. and Pirlo, G. | 2017 | Rejeitado (E2) |
| S92 | Comparison of t-test ranking with PCA and SEPCOR feature selection for wake and stage 1 sleep pattern recognition in multichannel electroencephalograms | Padma Shri, T.K. and Sriraam, N. | 2017 | Rejeitado (E2) |
| S93 | Hardware-Based Malware Detection Using Low-Level Architectural Features | Ozsoy, M. and Khasawneh, K.N. and Donovick, C. and Gorelik, I. and Abu-Ghazaleh, N. and Ponomarev, D. | 2016 | Rejeitado (E2) |
| S94 | Location based authentication scheme using BLE for high performance digital content management system | Choi, M. and Lee, J. and Kim, S. and Jeong, Y.-S. and Park, J.-H. | 2016 | Rejeitado (E2) |
| S95 | ODROID XU4 based implementation of decision level fusion approach for matching computer generated sketches | Fernandes, S.L. and Bala, G.J. | 2016 | Rejeitado (E1) |
| S96 | Merging permission and api features for android malware detection | Qiao, M. and Sung, A.H. and Liu, Q. | 2016 | Rejeitado (E2) |
| S98 | Analysis and methods to test classification of normal and pathological heart sound signals | Hendradi, R. and Arifin, A. and Shida, H. and Gunawan, S. and Purnomo, M.H. and Hasegawa, H. and Kanai, H. | 2016 | Rejeitado (E2) |
| S99 | Deeper knowledge tracing by modeling skill application context for better personalized learning | Huang, Y. | 2016 | Rejeitado (E2) |
| S100 | Skin artifact removal technique for breast cancer radar detection | Caorsi, S. and Lenzi, C. | 2016 | Rejeitado (E2) |
| S101 | Spectrum sensing in satellite cognitive radios: Blind signal detection technique | Khan, B.M. and Mustaqim, M. and Khawaja, B.A. and Shabeehulhusnain, S. | 2016 | Rejeitado (E2) |
| S102 | Performance comparison of texture feature analysis methods using PNN classifier for segmentation and classification of brain CT images | Padma, A. and Giridharan, N. | 2016 | Rejeitado (E2) |
| S103 | Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases | Janowczyk, A. and Madabhushi, A. | 2016 | Rejeitado (E2) |
| S104 | Joint unsupervised learning of deep representations and image clusters | Yang, J. and Parikh, D. and Batra, D. | 2016 | Rejeitado (E2) |
| S105 | Combining deep learning and survival analysis for asset health management | Liao, L. and Ahn, H.-I. | 2016 | Rejeitado (E2) |
| S110 | Using comparative behavior analysis to improve the impact of serious games on studentsâ€™ learning experience | Jaccard, D. and Hulaas, J. and Dumont, A. | 2016 | Rejeitado (E2) |
| S112 | Modeling skill combination patterns for deeper knowledge tracing | Huang, Y. and Guerra-Hollstein, J.D. and Brusilovsky, P. | 2016 | Rejeitado (E2) |
| S113 | A backpropagation neural network to improve arabic stemming | Mezher, K. and Omar, N. | 2015 | Rejeitado (E2) |
| S114 | A complete automatic short answer assessment system with student identification | Suwanwiwat, H. and Blumenstein, M. and Pal, U. | 2015 | Rejeitado (E1) |
| S115 | XFace: A Face Recognition System for Android Mobile Phones | Hu, J. and Peng, L. and Zheng, L. | 2015 | Rejeitado (E1) |
| S116 | LightLDA: Big topic models on modest computer clusters | Yuan, J. and Gao, F. and Ho, Q. and Dai, W. and Wei, J. and Zheng, X. and Xing, E.P. and Liu, T.-Y. and Ma, W.-Y. | 2015 | Rejeitado (E2) |
| S117 | Classification of test documents based on handwritten student ID's characteristics | Celar, S. and Stojkic, Z. and Seremet, Z. and Marusic, Z. and Zelenika, D. | 2015 | Rejeitado (E2) |
| S118 | A survey of fingerprint classification Part II: Experimental analysis and ensemble proposal | Galar, M. and Derrac, J. and Peralta, D. and Triguero, I. and Paternain, D. and Lopez-Molina, C. and GarcÃ­a, S. and BenÃ­tez, J.M. and Pagola, M. and Barrenechea, E. and Bustince, H. and Herrera, F. | 2015 | Rejeitado (E2) |
| S119 | A comparison of feature selection approach between greedy, IG-ratio, Chi-square, and mRMR in educational mining | Rachburee, N. and Punlumjeak, W. | 2015 | Rejeitado (E2) |
| S120 | An approach to extract roads from WorldView-2 satellite images: A case study from Istanbul-Turkey | Ogurlu, M. and Bayram, B. and Seker, D.Z. | 2015 | Rejeitado (E2) |
| S121 | A comparative study of feature selection techniques for classify student performance | Punlumjeak, W. and Rachburee, N. | 2015 | Rejeitado (E2) |
| S123 | Enculturating seamless language learning through artifact creation and social interaction process | Wong, L.-H. and Chai, C.S. and Aw, G.P. and King, R.B. | 2015 | Rejeitado (E2) |
| S124 | Automated diagnosis of Age-related Macular Degeneration using greyscale features from digital fundus images | Mookiah, M.R.K. and Acharya, U.R. and Koh, J.E.W. and Chandran, V. and Chua, C.K. and Tan, J.H. and Lim, C.M. and Ng, E.Y.K. and Noronha, K. and Tong, L. and Laude, A. | 2014 | Rejeitado (E2) |
| S125 | A flexible, interactive software tool for fitting the parameters of neuronal models | Friedrich, P. and Vella, M. and GulyÃ¡s, A.I. and Freund, T.F. and KÃ¡li, S. | 2014 | Rejeitado (E2) |
| S126 | Computer vision metrics: Survey, taxonomy, and analysis | Krig, S. | 2014 | Rejeitado (E2) |
| S128 | An Automatic English Composition scoring model based on neural network algorithm | Zhou, Y. and Fan, T. and Huang, G. | 2014 | Rejeitado (E2) |
| S129 | From model specification to simulation of biologically constrained networks of spiking neurons | Richmond, P. and Cope, A. and Gurney, K. and Allerton, D.J. | 2014 | Rejeitado (E2) |
| S130 | Off-line handwritten Thai name recognition for student identification in an automated assessment system | Suwanwiwat, H. and Nguyen, V. and Blumenstein, M. and Pal, U. | 2014 | Rejeitado (E2) |
| S131 | Off-Line Handwritten Bilingual Name Recognition for Student Identification in an Automated Assessment System | Suwanwiwat, H. and Nguyen, V. and Blumenstein, M. and Pal, U. | 2014 | Rejeitado (E2) |
| S132 | Factor analysis as the feature selection method in an emotion norm database | Wu, C.-H. and Kuo, B.-C. and Tzeng, G.-H. | 2014 | Rejeitado (E2) |
| S133 | Impact of classification of lab assignments and problem solving approach in object oriented programming lab course: A case study | Rajashekharaiah, K.M.M. and Patil, M.S. and Joshi, G.H. | 2014 | Rejeitado (E2) |
| S135 | 3D texture analysis of Solitary Pulmonary Nodules using co-occurrence matrix from volumetric lung CT images | Dhara, A.K. and Mukhopadhyay, S. and Khandelwal, N. | 2013 | Rejeitado (E2) |
| S136 | Face image classification using combined classifier | Sridhar, D. and Murali Krishna, Iv. | 2013 | Rejeitado (E2) |
| S137 | Designing an interactive storytelling game | Wu, C.-T. and Chung, S.-M. and Chang, S.-S. | 2012 | Rejeitado (E2) |
| S138 | Off-line restricted-set handwritten word recognition for student identification in a short answer question automated assessment system | Suwanwiwat, H. and Nguyen, V. and Blumenstein, M. | 2012 | Rejeitado (E2) |
| S140 | Application of neural network to brain-computer interface | Hsu, W.-Y. and Chiang, I.-J. | 2012 | Rejeitado (E2) |
| S141 | The USS Monitor: A pioneering model for education and outreach | Krop, D.S. and Holloway, A.G. and Alberg, D.W. | 2012 | Rejeitado (E1) |
| S142 | Automated screening of arrhythmia using wavelet based machine learning techniques | Martis, R.J. and Krishnan, M.M.R. and Chakraborty, C. and Pal, S. and Sarkar, D. and Mandana, K.M. and Ray, A.K. | 2012 | Rejeitado (E2) |
| S143 | Human features recognition with CNM: An applied study concerning undergraduate students | do Prado, H.A. and Ferneda, E. and Guadagnin, R. and dos Santos, G.M.F. | 2012 | Rejeitado (E2) |
| S144 | Research on improved K-means clustering algorithm | Zhang, Y. and Shan, H. and Li, J. and Zhou, J. | 2012 | Rejeitado (E1) |
| S145 | Fuzzy Hopfield neural network clustering for single-trial motor imagery EEG classification | Hsu, W.-Y. | 2012 | Rejeitado (E1) |
| S146 | Teaching and learning pattern recognition using neural networks | Tianxiang, B. and Li, Y.F. and Zhou, X. | 2011 | Rejeitado (E3) |
| S147 | SPKtool: An open source toolbox for electrophysiological data processing | Liu, X.-Q. and Wu, X. and Liu, C. | 2011 | Rejeitado (E3) |
| S148 | Extracting impact characteristics of sports training on EEG by Genetic Algorithm | Li, J. and Wang, W. | 2011 | Rejeitado (E3) |
| S149 | A new e-learning achievement evaluation model based on RBF-NN and similarity filter | Cheng, C.-H. and Chen, T.-L. and Wei, L.-Y. and Chen, J.-S. | 2011 | Rejeitado (E3) |
| S150 | Detection of relevance between long-term different professional training and brain development using EEG | Lili, F. and Wei, W. | 2011 | Rejeitado (E3) |
| S151 | Engineering competence? An interpretive investigation of engineering students' professional formation | Walther, J. and Kellam, N. and Sochacka, N. and Radcliffe, D. | 2011 | Rejeitado (E3) |
| S152 | Classification of heart sound based on S-Transform and neural network | Hadi, H.M. and Mashor, M.Y. and Suboh, M.Z. and Mohamed, M.S. | 2010 | Rejeitado (E3) |
| S153 | On the parallelization and optimization of the genetic-based ANN classifier for the diagnosis of students with learning disabilities | Wu, T.-K. and Huang, S.-C. and Lin, Y.-L. and Chang, H. and Meng, Y.-R. | 2010 | Rejeitado (E3) |
| S154 | Feature selection for efficient gender classification | Nazir, M. and Ishtiaq, M. and Batool, A. and Jaffar, M.A. and Mirza, A.M. | 2010 | Rejeitado (E3) |
| S155 | Tracing elementary school students' study tactic use in gStudy by examining a strategic and self-regulated learning | Malmberg, J. and JÃ¤rvenoja, H. and JÃ¤rvelÃ¤, S. | 2010 | Rejeitado (E3) |
| S156 | Image Processing and Pattern Recognition: Fundamentals and Techniques | Shih, F.Y. | 2010 | Rejeitado (E3) |
| S157 | Discover regulatory DNA elements using chromatin signatures and artificial neural network | Firpi, H.A. and Ucar, D. and Tan, K. | 2010 | Rejeitado (E3) |
| S158 | Optimal grouping by using genetic algorithm and support vector machines | Lin, K.-C. and Shiau, M.-L. and Lin, S.-Y. and Tai, J. | 2009 | Rejeitado (E3) |
| S159 | Advanced biosignal processing | NaÃ¯t-Ali, A. | 2009 | Rejeitado (E3) |
| S161 | KPCA SVM with GA model for technological achievements of college students forecasting | Zhao, J. and Song, Z. and Jiao, C. | 2009 | Rejeitado (E3) |
| S163 | Emotional agent in serious game (DINO) | Zhang, H.L. and Shen, Z. and Tao, X. and Miao, C. and Li, B. and Ailiya, A. and Cai, Y. | 2009 | Rejeitado (E3) |
| S164 | On capturing malware dynamics in mobile power-law networks | Bose, A. and Shin, K.G. | 2008 | Rejeitado (E3) |
| S165 | Conditional information and information loss for flexible feature extraction | Kamimura, R. | 2008 | Rejeitado (E3) |
| S166 | Modeling network attacks for scenario construction | Al-Mamory, S.O. and Zhang, H. and Abbas, A.R. | 2008 | Rejeitado (E3) |
| S167 | Evaluation of ANN and SVM classifiers as predictors to the diagnosis of students with learning disabilities | Wu, T.-K. and Huang, S.-C. and Meng, Y.-R. | 2008 | Rejeitado (E3) |
| S175 | A technique to reduce the test case suites for regression testing based on a self-organizing neural network architecture | Da Silva SimÃ£o, A. and De Mello, R.F. and Senger, L.J. | 2006 | Rejeitado (E3) |
| S178 | Effects of feature selection on the identification of students with learning disabilities using ANN | Wu, T.-K. and Huang, S.-C. and Meng, Y.-R. | 2006 | Rejeitado (E3) |
| S179 | Teseo: A vectoriser of historical seismograms | Pintore, S. and Quintiliani, M. and Franceschi, D. | 2005 | Rejeitado (E3) |
| S180 | Recognition and translation of the Myanmar printed text based on hopfield neural network | Swe, T. and Tin, P. | 2005 | Rejeitado (E3) |
| S181 | Feature recognition in solar images | Zharkova, V. and Ipson, S. and Benkhalil, A. and Zharkov, S. | 2005 | Rejeitado (E3) |
| S182 | ATR Workbench for Automating Image Analysis | English, R.A. and Rawlinson, S.J. and Sandirasegaram, N.M. | 2003 | Rejeitado (E3) |
| S183 | Question classification for e-learning by artificial neural network | Fei, T. and Heng, W.J. and Toh, K.C. and Qi, T. | 2003 | Rejeitado (E3) |
| S184 | The advantage of industrial R&D in the training of university students | Fulcher, J.A. | 2001 | Rejeitado (E3) |
| S186 | Speech visualization by integrating features for the hearing impaired | Watanabe, A. | 2000 | Rejeitado (E3) |
| S187 | Machine translation-based bug localization technique for bridging lexical gap | Xiao, Y. and Keung, J. and Bennin, K.E. and Mi, Q. | 2018 | Duplicado |
| S188 | Cross-Project Transfer Representation Learning for Vulnerable Function Discovery | Lin, G. and Zhang, J. and Luo, W. and Pan, L. and Xiang, Y. and De Vel, O. and Montague, P. | 2018 | Duplicado |
| S191 | Electronic proofs in mathematics education - A South African Teacher Professional Development (TPD) course informing the conceptualisation of an e-proof system authoring support workshop | Platz, M. and Krieger, M. and Niehaus, E. and Winter, K. | 2017 | Duplicado |
| S192 | CCLearner: A deep learning-based clone detection approach | Li, L. and Feng, H. and Zhuang, W. and Meng, N. and Ryder, B. | 2017 | Duplicado |
| S193 | Long short-term memory RNN for biomedical named entity recognition | Lyu, C. and Chen, B. and Ren, Y. and Ji, D. | 2017 | Duplicado |
| S194 | Poster: Vulnerability discovery with function representation learning from unlabeled projects | Lin, G. and Zhang, J. and Luo, W. and Pan, L. and Xiang, Y. | 2017 | Duplicado |
| S195 | Source code classification using Neural Networks | Gilda, S. | 2017 | Duplicado |
| S196 | Applying deep learning based automatic bug triager to industrial projects | Lee, S.-R. and Heo, M.-J. and Lee, C.-G. and Kim, M. and Jeong, G. | 2017 | Duplicado |
| S197 | Software defect prediction via convolutional neural network | Li, J. and He, P. and Zhu, J. and Lyu, M.R. | 2017 | Duplicado |
| S198 | Semantically Enhanced Software Traceability Using Deep Learning Techniques | Guo, J. and Cheng, J. and Cleland-Huang, J. | 2017 | Duplicado |
| S199 | Replicating Parser Behavior Using Neural Machine Translation | Alexandru, C.V. and Panichella, S. and Gall, H.C. | 2017 | Duplicado |
| S200 | A memory-Augmented neural model for automated grading | Zhao, S. and Zhang, Y. and Xiong, X. and Botelho, A. and Heffernan, N. | 2017 | Duplicado |
| S201 | A hybrid scheme for automated essay grading based on LVQ and NLP techniques | Shehab, A. and Elhoseny, M. and Hassanien, A.E. | 2017 | Duplicado |
| S202 | Character-Level Convolutional Networks for Arithmetic Operator Character Recognition | Liang, Z. and Li, Q. and Liao, S. | 2017 | Duplicado |
| S203 | DeepFix: Fixing Common C language errors by deep learning | Gupta, R. and Pal, S. and Kanade, A. and Shevade, S. | 2017 | Duplicado |
| S204 | SibStCNN and TBCNN + kNN-TED: New models over tree structures for source code classification | Phan, A.V. and Nguyen, M.L. and Bui, L.T. | 2017 | Duplicado |
| S205 | Source code authorship attribution using long short-term memory based networks | Alsulami, B. and Dauber, E. and Harang, R. and Mancoridis, S. and Greenstadt, R. | 2017 | Duplicado |
| S207 | AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP | Soliman, A.B. and Eissa, K. and El-Beltagy, S.R. | 2017 | Duplicado |
| S209 | A Comparison among ARIMA, BP-NN, and Moga-NN for Software Clone Evolution Prediction | Pati, J. and Kumar, B. and Manjhi, D. and Shukla, K.K. | 2017 | Duplicado |
| S211 | Sk-P: A neural program corrector for MOOCs | Pu, Y. and Narasimhan, K. and Solar-Lezama, A. and Barzilay, R. | 2016 | Duplicado |
| S212 | Deep learning code fragments for code clone detection | White, M. and Tufano, M. and Vendome, C. and Poshyvanyk, D. | 2016 | Duplicado |
| S213 | Automatically learning semantic features for defect prediction | Wang, S. and Liu, T. and Tan, L. | 2016 | Duplicado |
| S214 | Convolutional neural networks over tree structures for programming language processing | Mou, L. and Li, G. and Zhang, L. and Wang, T. and Jin, Z. | 2016 | Duplicado |
| S215 | Bot.zen @ EVALITA 2016 - A minimally-deep learning PoS-tagger (trained for Italian Tweets) | Stemle, E.W. | 2016 | Duplicado |
| S216 | A convolutional attention network for extreme summarization of source code | Allamanis, M. and Peng, H. and Sutton, C. | 2016 | Duplicado |
| S225 | Research on the emotion recognition based on the fuzzy neural network in the intelligence education system | Pan, X. | 2011 | Rejeitado (E3) |
| S226 | Real-time binaural sound source localization using sparse coding and SOM | Hwang, D.-H. and Choi, J.-S. | 2010 | Rejeitado (E3) |
| S227 | Robust workflow recognition using holistic features and outlier-tolerant fused hidden markov models | Voulodimos, A. and Grabner, H. and Kosmopoulos, D. and Van Gool, L. and Varvarigou, T. | 2010 | Rejeitado (E3) |
| S228 | BioCASE: Accelerating software development of genome-wide filtering applications | Montes, R. and Abad-Grau, M.M. | 2009 | Rejeitado (E3) |
| S229 | Jawi character speech-to-text engine using linear predictive and neural network for effective reading | Othman, Z.A. and Razak, Z. and Abdullah, N.A. and Yakub, M. and Mohd. Yusoff, Z.B. | 2009 | Rejeitado (E3) |
| S230 | Evaluation system for e-learning with pattern mining tools | Ribeiro, B. and Cardoso, A. | 2008 | Rejeitado (E3) |
| S231 | Fingerprint recognition using the fuzzy Sugeno integral for response integration in modular neural networks | Melin, P. and Bravo, D. and Castillo, O. | 2008 | Rejeitado (E3) |
| S232 | Automatic classification of spinal deformity by using four symmetrical features on the moire images | Kim, H. and Nakano, S. and Tan, J.K. and Ishikawa, S. and Otsuka, Y. and Shimizu, H. and Shinomiya, T. | 2007 | Rejeitado (E3) |
| S233 | A study on RGB color extraction of psoriasis lesion using Principle Component Analysis (PCA) | Hashim, H. and Rahman, R.A. and Jarmin, R. and Taib, M.N. | 2006 | Rejeitado (E3) |
| S234 | MeaBench: A toolset for multi-electrode data acquisition and on-line analysis | Wagenaar, D. and Demarse, T.B. and Potter, S.M. | 2005 | Rejeitado (E3) |
| S235 | EMG-control of prostheses by switch signals: Extraction and classification of features | Reischl, M. and GrÃ¶ll, L. and Mikut, R. | 2004 | Rejeitado (E3) |
| S236 | A Turkish handprint character recognition system | Ã‡apar, A. and TaÅŸdemir, K. and Kilic, Ã–. and GÃ¶kmen, M. | 2003 | Rejeitado (E3) |
| S237 | CSIDC: Competing students design real-world systems | Clements, A. | 2003 | Rejeitado (E3) |
| S238 | Comparison of linear predictive analysis methods for ANN-based speaker identification | Inal, M. and BÃ¼tÃ¼n, E. and Erkan, K. and Yildirum, M. and Ã‡eken, C. | 2000 | Rejeitado (E3) |
| S239 | ARGUS: an automated multi-agent visitor identification system | Sukthankar, Rahul and Stockton, Robert G. | 1999 | Rejeitado (E3) |
| S240 | From brains to neural nets to brains | Harth, E. | 1997 | Rejeitado (E3) |
| S241 | Design of a handprint recognition system | Garris, M.D. and Blue, J.L. and Candela, G.T. and Dimmick, D.L. and Geist, J. and Grother, P.J. and Janet, S.A. and Omidvar, O.M. and Wilson, C.L. | 1997 | Rejeitado (E3) |
| S242 | Public domain optical character recognition | Garris, Michael D. and Blue, James L. and Candela, Gerald T. and Dimmick, Darrin L. and Geist, Jon C. and Grother, Patrick J. and Janet, Stanley A. and Wilson, Charles L. | 1995 | Rejeitado (E3) |
| S243 | Classiwng Cells for Cancer Diagnosis Using Neural Netwvorks | Moallemi, C. | 1991 | Rejeitado (E3) |

Tabela 4 - Seleção Preliminar da SCOPUS

## Web Of Science

**Data de busca**: 28 de abril, 07 de maio de 2018

**String de busca**: ( "neural network" OR "recurrent neural networks" OR "rnn" OR "long short-term memory" OR "lstm" OR "convolutional neural network" OR "cnn" OR "deep learning" ) AND ( "automated grading" OR "automatic grading" OR "program repair" OR "code repair" OR "corrector" OR "fault localization" OR "programming errors" OR "word embedding" OR "feature extraction" OR "representation learning" OR "syntax" OR "token" OR "trace" ) AND ( "programming education" OR "computer science education" OR "massive open online course" OR "mooc" OR "introductory programming" OR "student" OR "big code" OR "open source" OR "source code" OR "program analysis" )

**Campos pesquisados**: Título, Palavras-chaves e Resumo

**Período considerado**: 2012 a maio de 2018

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID Paper | Title | Authors | Year | Status/Selection |
| W1 | Research on Improved K-means Clustering Algorithm | Zhang, Yinsheng and Shan, Huilin and Li, Jiaqiang and Zhou, Jie | 2012 | Rejeitado (E1) |
| W2 | Authorship attribution of source code by using back propagation neuralnetwork based on particle swarm optimization | Yang, Xinyu and Xu, Guoai and Li, Qi and Guo, Yanhui and Zhang, Miao | 2017 | Aceito (I1, I2, I3, I4) |
| W3 | Implicit Heterogeneous Features Embedding in Deep Knowledge Tracing | Yang, Haiqin and Cheung, Lap Pong | 2018 | Duplicado |
| W4 | An Emotion Recognition Model Based on Facial Recognition in VirtualLearning Environment | Yang, D. and Alsadoon, Abeer and Prasad, P. W. C. and Singh, A. K. andElchouemi, A. | 2018 | Rejeitado (E2) |
| W5 | Improving Bug Localization with an Enhanced Convolutional Neural Network | Xiao, Yan and Keung, Jacky and Mi, Qing and Bennin, Kwabena E. | 2017 | Aceito (I1, I2, I3, I4) |
| W6 | Deep Learning Code Fragments for Code Clone Detection | White, Martin and Tufano, Michele and Vendome, Christopher andPoshyvanyk, Denys | 2016 | Duplicado |
| W7 | Automatically Learning Semantic Features for Defect Prediction | Wang, Song and Liu, Taiyue and Tan, Lin | 2016 | Duplicado |
| W8 | ADAGE signature analysis: differential expression analysis withdata-defined gene sets | Tan, Jie and Huyck, Matthew and Hu, Dongbo and Zelaya, Rene A. andHogan, Deborah A. and Greene, Casey S. | 2017 | Rejeitado (E2) |
| W9 | A Hybrid Scheme for Automated Essay Grading Based on LVQ and NLPTechniques | Shehab, Abdulaziz and Elhoseny, Mohamed and Hassanien, Aboul Ella | 2016 | Rejeitado (E2) |
| W10 | Improvements in IITG Assamese Spoken Query System: Background NoiseSuppression and Alternate Acoustic Modeling | Shahnawazuddin, S. and Thotappa, Deepak and Dey, Abhishek and Imani,Siddika and Prasanna, S. R. M. and Sinha, Rohit | 2017 | Rejeitado (E2) |
| W11 | Bad Teacher or Unruly Student: Can Deep Learning Say Something in ImageForensics Analysis? | Rota, Paolo and Sangineto, Enver and Conotter, Valentina andPramerdorfer, Christopher | 2016 | Rejeitado (E2) |
| W12 | From Model Specification to Simulation of Biologically ConstrainedNetworks of Spiking Neurons | Richmond, Paul and Cope, Alex and Gurney, Kevin and Allerton, David J. | 2014 | Rejeitado (E2) |
| W13 | Impact of classification of lab assignments and problem solving approachin Object Oriented Programming Lab course: A case study | Rajashekharaiah, K. M. M. and Patil, Mahesh S. and Joshi, G. H. | 2014 | Rejeitado (E2) |
| W14 | sk\\_p: A Neural Program Corrector for MOOCs | Pu, Yewen and Narasimhan, Karthik and Solar-Lezama, Armando andBarzilay, Regina | 2016 | Duplicado |
| W15 | A Comparison Among ARIMA, BP-NN, and MOGA-NN for Software CloneEvolution Prediction | Pati, Jayadeep and Kumar, Babloo and Manjhi, Devesh and Shukla, K. K. | 2017 | Duplicado |
| W16 | Segmentation and Classification of Brain CT Images Using CombinedWavelet Statistical Texture Features | Padma, A. and Sukanesh, R. | 2014 | Rejeitado (E2) |
| W17 | An open access database for the evaluation of heart sound algorithms | Liu, Chengyu and Springer, David and Li, Qiao and Moody, Benjamin andJuan, Ricardo Abad and Chorro, Francisco J. and Castells, Francisco andRoig, Jose Millet and Silva, Ikaro and Johnson, Alistair E. W. and Syed,Zeeshan and Schmidt, Samuel E. and Papadaniil, Chrysa D. andHadjileontiadis, Leontios and Naseri, Hosein and Moukadem, Ali andDieterlen, Alain and Brandt, Christian and Tang, Hong and Samieinasab,Maryam and Samieinasab, Mohammad Reza and Sameni, Reza and Mark, RogerG. and Clifford, Gari D. | 2016 | Rejeitado (E1) |
| W18 | CCLearner: A Deep Learning-Based Clone Detection Approach | Li, Liuqing and Feng, He and Zhuang, Wenjie and Meng, Na and Ryder,Barbara | 2017 | Duplicado |
| W19 | Software Defect Prediction via Convolutional Neural Network | Li, Jian and He, Pinjia and Zhu, Jieming and Lyu, Michael R. | 2017 | Duplicado |
| W20 | Applying Deep Learning Based Automatic Bug Triager to IndustrialProjects | Lee, Sun-Ro and Heo, Min-Jae and Lee, Chan-Gun and Kim, Milhan andJeong, Gaeul | 2017 | Duplicado |
| W21 | How deep learning extracts and learns leaf features for plantclassification | Lee, Sue Han and Chan, Chee Seng and Mayo, Simon Joseph and Remagnino,Paolo | 2017 | Rejeitado (E2) |
| W22 | Detection of Breast Cancer from Histopathology image and ClassifyingBenign and Malignant State Using Fuzzy Logic | Johra, Fatema-Tuz and Shuvo, Md Maruf Hossain | 2016 | Rejeitado (E2) |
| W23 | Using Comparative Behavior Analysis to Improve the Impact of SeriousGames on Students' Learning Experience | Jaccard, Dominique and Hulaas, Jarle and Dumont, Ariane | 2016 | Rejeitado (E2) |
| W24 | XFace: A Face Recognition System for Android Mobile Phones | Hu, Jiawei and Peng, Liangrui and Zheng, Li | 2015 | Duplicado |
| W25 | Application of Neural Network to Brain-Computer Interface | Hsu, Wei-Yen and Chiang, I-Jen | 2012 | Rejeitado (E2) |
| W26 | SINGLE-TRIAL MOTOR IMAGERY CLASSIFICATION USING ASYMMETRY RATIO, PHASERELATION, WAVELET-BASED FRACTAL, AND THEIR SELECTED COMBINATION | Hsu, Wei-Yen | 2013 | Rejeitado (E1) |
| W27 | Enhanced Active Segment Selection for Single-Trial EEG Classification | Hsu, Wei-Yen | 2012 | Rejeitado (E2) |
| W28 | NONLINEAR ANALYSIS OF CORONARY ARTERY DISEASE, MYOCARDIAL INFARCTION,AND NORMAL ECG SIGNALS | Hagiwara, Yuki and Faust, Oliver | 2017 | Rejeitado (E2) |
| W29 | Semantically Enhanced Software Traceability Using Deep LearningTechniques | Guo, Jin and Cheng, Jinghui and Cleland-Huang, Jane | 2017 | Aceito (I1, I2, I3, I4) |
| W30 | Source Code Classification Using Neural Networks | Gilda, Shlok | 2017 | Duplicado |
| W31 | NiftyNet: a deep-learning platform for medical imaging | Gibson, Eli and Li, Wenqi and Sudre, Carole and Fidon, Lucas and Shakir,Dzhoshkun I. and Wang, Guotai and Eaton-Rosen, Zach and Gray, Robert andDoel, Tom and Hu, Yipeng and Whyntie, Tom and Nachev, Parashkev andModat, Marc and Barratt, Dean C. and Ourselin, Sebastien and Cardoso, M.Jorge and Vercauteren, Tom | 2018 | Duplicado |
| W32 | ODROID XU4 based implementation of decision level fusion approach formatching computer generated sketches | Fernandes, Steven Lawrence and Bala, G. Josemin | 2016 | Rejeitado (E2) |
| W33 | Classification of Test Documents Based on Handwritten Student ID'sCharacteristics | Celar, Stipe and Stojkic, Zeljko and Seremet, Zeljko and Marusic, Zeljkoand Zelenika, Danijel | 2015 | Rejeitado (E2) |
| W34 | Skin artifact removal technique for breast cancer radar detection | Caorsi, S. and Lenzi, C. | 2016 | Duplicado |
| W35 | A Facial Recognition Application Based on Incremental SupervisedLearning | Buhus, Elena Roxana and Grama, Lacrimioara and Serbu, Catalina | 2017 | Rejeitado (E2) |
| W36 | Bug Localization with Combination of Deep Learning and InformationRetrieval | An Ngoc Lam and Anh Tuan Nguyen and Hoan Anh Nguyen and Nguyen, Tien N. | 2017 | Duplicado |
| W37 | Convolutional Neural Networks on Assembly Code for Predicting SoftwareDefects | Anh Viet Phan and Minh Le Nguyen | 2017 | Aceito (I1, I2, I3, I4) |
| W38 | Replicating Parser Behavior using Neural Machine Translation | Alexandra, Carol V. and Panichella, Sebastiano and Gall, Harald C. | 2017 | Duplicado |

Tabela 5 - Seleção preliminar da Web Of Science

## IEEE

**Data de busca**: 07 de maio de 2018

**String de busca**: ( "neural network" OR "recurrent neural networks" OR "rnn" OR "long short-term memory" OR "lstm" OR "convolutional neural network" OR "cnn" OR "deep learning" ) AND ( "automated grading" OR "automatic grading" OR "program repair" OR "code repair" OR "corrector" OR "fault localization" OR "programming errors" OR "word embedding" OR "feature extraction" OR "representation learning" OR "syntax" OR "token" OR "trace" ) AND ( "programming education" OR "computer science education" OR "massive open online course" OR "mooc" OR "introductory programming" OR "student" OR "big code" OR "open source" OR "source code" OR "program analysis" )

**Campos pesquisados**: Título, Palavras-chaves e Resumo

**Período considerado**: 2012 a maio de 2018

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID Paper | Title | Authors | Year | Status/Selection |
| I1 | Automatically Learning Semantic Features for Defect Prediction | S. Wang and T. Liu and L. Tan | 2016 | Duplicado |
| I2 | Deep learning code fragments for code clone detection | M. White and M. Tufano and C. Vendome and D. Poshyvanyk | 2016 | Duplicado |
| I3 | Toward Deep Learning Software Repositories | M. White and C. Vendome and M. Linares-Vasquez and D. Poshyvanyk | 2015 | Duplicado |
| I4 | CCLearner: A Deep Learning-Based Clone Detection Approach | L. Li and H. Feng and W. Zhuang and N. Meng and B. Ryder | 2017 | Duplicado |
| I5 | Semantically Enhanced Software Traceability Using Deep Learning Techniques | J. Guo and J. Cheng and J. Cleland-Huang | 2017 | Duplicado |
| I6 | Syntax and sensibility: Using language models to detect and correct syntax errors | E. A. Santos and J. C. Campbell and D. Patel and A. Hindle and J. N. Amaral | 2018 | Aceito (I1, I2, I3, I4) |
| I7 | Replicating Parser Behavior Using Neural Machine Translation | C. V. Alexandru and S. Panichella and H. C. Gall | 2017 | Duplicado |
| I8 | Convolutional neural networks on assembly code for predicting software defects | A. V. Phan and M. Le Nguyen | 2017 | Duplicado |
| I9 | TENCON '89. Fourth IEEE Region 10 International Conference. 'Information Technologies for the 90's' E/sup 2/C/sup 2/; Energy, Electronics, Computers, Communications (Cat. No.A89CH2766-4) | | 1989 | Rejeitado (E4) |

Tabela 6 - Seleção preliminar da IEEE

## ACM

**Data de busca**: 07 de maio de 2018

**String de busca**: ( "neural network" OR "recurrent neural networks" OR "rnn" OR "long short-term memory" OR "lstm" OR "convolutional neural network" OR "cnn" OR "deep learning" ) AND ( "automated grading" OR "automatic grading" OR "program repair" OR "code repair" OR "corrector" OR "fault localization" OR "programming errors" OR "word embedding" OR "feature extraction" OR "representation learning" OR "syntax" OR "token" OR "trace" ) AND ( "programming education" OR "computer science education" OR "massive open online course" OR "mooc" OR "introductory programming" OR "student" OR "big code" OR "open source" OR "source code" OR "program analysis" )

**Campos pesquisados**: Título, Palavras-chaves e Resumo

**Período considerado**: 2012 a maio de 2018

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID Paper | Title | Authors | Year | Status/Selection |
| A1 | A Memory-Augmented Neural Model for Automated Grading | Zhao, Siyuan and Zhang, Yaqiong and Xiong, Xiaolu and Botelho, Anthony and Heffernan, Neil | | Duplicado |
| A2 | Applying Deep Learning Based Automatic Bug Triager to Industrial Projects | Lee, Sun-Ro and Heo, Min-Jae and Lee, Chan-Gun and Kim, Milhan and Jeong, Gaeul | 2017 | Duplicado |
| A3 | A Survey of Data Mining Techniques for Malware Detection Using File Features | Siddiqui, Muazzam and Wang, Morgan C. and Lee, Joohan | 2008 | Duplicado |
| A4 | Automatically Learning Semantic Features for Defect Prediction | Wang, Song and Liu, Taiyue and Tan, Lin | 2016 | Duplicado |
| A5 | BlazingText: Scaling and Accelerating Word2Vec Using Multiple GPUs | Gupta, Saurabh and Khare, Vineet | 2017 | Duplicado |
| A6 | Bug Localization with Combination of Deep Learning and Information Retrieval | Lam, An Ngoc and Nguyen, Anh Tuan and Nguyen, Hoan Anh and Nguyen, Tien N. | 2017 | Duplicado |
| A7 | Deeper Knowledge Tracing by Modeling Skill Application Context for Better Personalized Learning | Huang, Yun | 2016 | Duplicado |
| A8 | Deep Knowledge Tracing On Programming Exercises | Wang, Lisa and Sy, Angela and Liu, Larry and Piech, Chris | | Duplicado |
| A9 | Deep Learning Code Fragments for Code Clone Detection | White, Martin and Tufano, Michele and Vendome, Christopher and Poshyvanyk, Denys | 2016 | Duplicado |
| A10 | Deep Model for Dropout Prediction in MOOCs | Wang, Wei and Yu, Han and Miao, Chunyan | 2017 | Duplicado |
| A11 | Deep Unsupervised Feature Learning for Natural Language Processing | Gouws, Stephan | 2012 | Duplicado |
| A12 | Dynamic Key-Value Memory Networks for Knowledge Tracing | Zhang, Jiani and Shi, Xingjian and King, Irwin and Yeung, Dit-Yan | 2017 | Duplicado |
| A13 | Evaluating Neural Networks As a Method for Identifying Students in Need of Assistance | Castro-Wunsch, Karo and Ahadi, Alireza and Petersen, Andrew | 2017 | Duplicado |
| A14 | Guided Code Synthesis Using Deep Neural Networks | Alexandru, Carol V. | 2016 | Duplicado |
| A15 | Incorporating Rich Features into Deep Knowledge Tracing | Zhang, Liang and Xiong, Xiaolu and Zhao, Siyuan and Botelho, Anthony and Heffernan, Neil T. | | Duplicado |
| A16 | Intelligent Intrusion Detection System Using a Committee of Experts | Subramanian, Krishnan and Senthilkumar, Sachin and Thiagarajan, Balasubramanian | 2016 | Duplicado |
| A17 | LightLDA: Big Topic Models on Modest Computer Clusters | Yuan, Jinhui and Gao, Fei and Ho, Qirong and Dai, Wei and Wei, Jinliang and Zheng, Xun and Xing, Eric Po and Liu, Tie-Yan and Ma, Wei-Ying | 2015 | Duplicado |
| A18 | Modular Array-based GPU Computing in a Dynamically-typed Language | Springer, Matthias and Wauligmann, Peter and Masuhara, Hidehiko | 2017 | Duplicado |
| A19 | Personalized Adaptive Learning Using Neural Networks | Chaplot, Devendra Singh and Rhim, Eunhee and Kim, Jihie | | Duplicado |
| A20 | POSTER: Vulnerability Discovery with Function Representation Learning from Unlabeled Projects | Lin, Guanjun and Zhang, Jun and Luo, Wei and Pan, Lei and Xiang, Yang | 2017 | Duplicado |
| A21 | Predicting Vulnerable Software Components Through Deep Neural Network | Pang, Yulei and Xue, Xiaozhen and Wang, Huaying | 2017 | Duplicado |
| A22 | Replicating Parser Behavior Using Neural Machine Translation | Alexandru, Carol V. and Panichella, Sebastiano and Gall, Harald C. | 2017 | Duplicado |
| A23 | Semantically Enhanced Software Traceability Using Deep Learning Techniques | Guo, Jin and Cheng, Jinghui and Cleland-Huang, Jane | 2017 | Duplicado |
| A24 | Sk\_P: A Neural Program Corrector for MOOCs | Pu, Yewen and Narasimhan, Karthik and Solar-Lezama, Armando and Barzilay, Regina | 2016 | Duplicado |
| A25 | Toward Deep Learning Software Repositories | White, Martin and Vendome, Christopher and Linares-V\'{a}squez, Mario and Poshyvanyk, Denys | 2015 | Duplicado |

Tabela 7 - Seleção preliminar da ACM