O Uso de Inteligência Artificial em Testes de Software: Um Mapeamento Sistemático Quantitativo

Yann Pereira da Silva(yann.pereira@dcx.ufpb.br)
Rafael Marrocos Magalhães(rafael@dcx.ufpb.br)

[1]Campus IV - Universidade Federal da Paraíba (UFPB) - Rio Tinto - PB - Brasil

Resumo: Este artigo trata da condução de um estudo de mapeamento sistemático quantitativo que investiga uso de técnicas de inteligência artificial (IA) e de aprendizado de máquina (AM) (do inglês *machine learning* - ML) aplicadas na área de testes de softwares no contexto de engenharia de software. Foram considerados estudos primários e secundários oriundos de fontes revisadas por pares e de motores de buscas reconhecidos e acessíveis pela comunidade acadêmica. O trabalho foca essencialmente na temporalidade (5 anos mais recentes) dos estudos, de tal forma a captar o momento presente das pesquisas atuais a respeito da combinação destas duas áreas, numa perspectiva prioritariamente quantitativa.

Palavras-chaves: Mapeamento Sistemático, Testes de Softwares, Inteligência Artificial, Machine Learning

1. Introdução

Segmento de estudo das ciências da computação, a Inteligência Artificial (IA), segue em crescente volume de pesquisa e desenvolvimento a partir de diferentes frentes. Jordan e Mitchell(JORDAN; MITCHELL, 2015)**,** declaram que a IA e a área de aprendizagem de máquina (AM) são, na atualidade, "um dos campos técnicos que mais cresce, concentrando-se na intersecção entre a ciência da computação e a estatística", destacam ainda, que a adoção de métodos de IA e AM podem ser encontrada em segmentos como "ciência, tecnologia, comércio e em áreas como engenharia, saúde, manufatura, educação, finanças, políticas", dentre outras.

Especificamente no âmbito da tecnologia da informação, a IA também está presente nos projetos de desenvolvimento de softwares de grandes empresas, incluindo os processos de testes de software. Estes processos, por consequência da interação com o uso de IA, se alavancam em diferentes técnicas e práticas, e é exatamente neste contexto que esta pesquisa se insere.

No domínio dos testes de software já é possível constatar aplicações de IA em processos de automação, criação de conteúdo e simulação de uso em testes de softwares. Em um exemplo de aplicação recente, Ramos(RAMOS,2020) demonstra o uso de IA para fazer predição de defeitos no momento de realização de *commits* de repositórios de softwares, com a intenção de detectar possíveis falhas de software.

No final deste artigo veremos que mesmo com pesquisas em volume crescente sobre a inclusão de técnicas de Inteligência Artificial nos testes de software ou em seus processos, ainda não há um método específico ou um protocolo amplamente adotado para a variedade de abordagens e possibilidades existentes. E é sustentado nessa ausência de processos ou práticas padrões que este estudo se motiva. Esse estudo tem como intenção primária averiguar, quantitativamente, os esforços nos cinco(5) anos mais recentes que tratam da

[1] Trabalho de conclusão de curso, sob orientação do professor Rafael Marrocos Magalhães submetido ao Curso de Bacharelado em Sistemas de Informação do Centro de Ciências Aplicadas e Educação (CCAE) da Universidade Federal da Paraíba, como parte dos requisitos necessários para obtenção do grau de BACHAREL EM SISTEMAS DE INFORMAÇÃO.

aplicação de técnicas de Inteligência Artificial nas diferentes áreas e escopos do segmento de Teste de Softwares. Com isso, intenta-se subsidiar uma compreensão quantitativa sobre o que está sendo pesquisado e desenvolvido na interseção dessas duas grandes áreas.

O texto segue através de mais quatro seções sendo a próxima sobre os aspectos teóricos mais relevantes das áreas de Testes de Softwares e Inteligência Artificial. Na seção 3, de metodologia, é definido o protocolo utilizado para realizar este estudo bem como a descrição de sua condução. A seção 4 abriga os resultados obtidos a partir da análise dos dados obtidos e, em especial, sua descrição como respostas às questões de pesquisa definidas na seção 3. O texto se encerra na seção 5 com as considerações finais.

2. Fundamentação Teórica

Considerando-se a aplicação de IA em Testes de Softwares e por situar-se entre dois campos de saberes distintos, optou-se por incluir uma breve descrição dos principais pontos de cada uma delas nesta seção, conforme segue o texto. Desta forma fica mais acessível para leitores especialistas dos diferentes segmentos.

2.1 Testes de Software

O processo de testes é uma etapa padrão das atividades da Engenharia de Software e é aplicada para garantir que o software seja entregue com a melhor qualidade possível. Dentro desse grande empreendimento, que é a aplicação de teste, existem outras subtarefas nos quais encontram-se diversas ferramentas e técnicas otimizadas em geral para economizar tempo sem perder a qualidade.

Sommerville (SOMMERVILLE, 2019) descreve um fluxo de uma instância de aplicação de testes como prática do desenvolvimento de softwares que considera quatro passos (ver Figura 1), a saber: projetar casos de teste, preparar, executar, comprar e relatar os testes. Entretanto, numa perspectiva de mais alto nível, Olsen et al (OLSEN et al., 2018), definem que as fases dos testes de software consistem nas etapas de Planejamento, Monitoramento e Controle, Análise, Modelagem, Implementação, Execução e Conclusão. Baseado nessa perspectiva, descreve-se a seguir cada uma delas.

Na etapa de planejamento encontram-se estimativas, riscos, premissas, requisitos a serem testados, métricas, etc. Nessa fase, as estratégias de testes são definidas e todos os testes são planejados. Em monitoramento e controle, os testes são monitorados e avaliados de acordo com as métricas definidas no plano de teste da etapa de planejamento, garantindo com que o teste siga fielmente as métricas definidas. É imprescindível que todos os registros desta etapa sejam comunicados aos *stakeholders*.

Na terceira etapa, de análise, existem várias atividades principais, mas em resumo, nesta etapa são identificadas e definidas as condições de teste e os recursos testáveis da aplicação. A seguinte, de modelagem, as condições de testes são descritas de forma mais interativas, formando assim os famosos casos de teste que se pode assimilar a um roteiro onde o passo a passo dos testes são descritos, se na fase de análise é definido o que deve ser testado, na fase de modelagem é definido como esses recursos

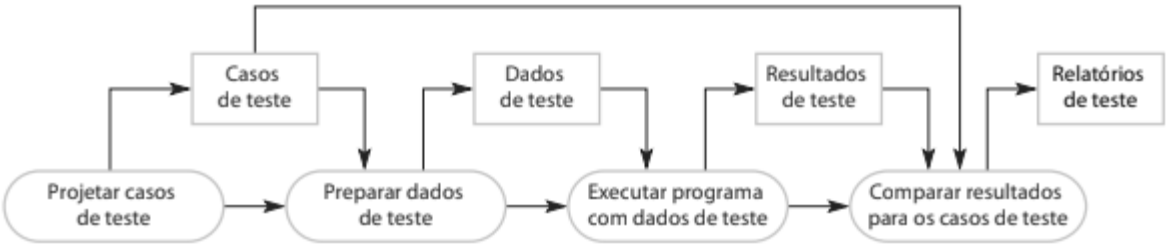

Figura 1 - Imagem replicada do livro "Engenharia de software" - SOMMERVILLE, I. Engenharia de Software. 10ª. Ed. São Paulo: Pearson Prentice Hall, 2019 - Capítulo 8.

devem ser testados. Na etapa de implementação, todo o ambiente é preparado para que os testes sejam executados, caso sejam realizados testes automatizados é nesta fase que os scripts são preparados. Sendo subsequente a fase de modelagem, onde é definido como os recursos devem ser testados, na fase de implementação é assegurado que os testes tenham todos os recursos necessários para serem executados.

Na etapa de execução, os testes são executados de acordo como foram programados, sejam eles de forma manual ou usando ferramentas de automação, é importante sempre comparar os resultados obtidos com os resultados esperados, todo resultado deverá ser registrado. Por fim, na etapa de conclusão todas as evidências dos testes são coletadas, o testador cria um documento com o resumo dos testes para ser entregue aos stakeholders. A partir desses documentos, os resultados serão analisados e usados para melhorar o processo de testes.

2.2 Inteligência Artificial

Rouhiainen (ROUHIAINEN, 2018) define a inteligência artificial como a capacidade das máquinas de aprender com os dados treinados em algoritmos e usar o que aprendeu para tomar decisões como um ser humano.

A inteligência artificial possui grandes subáreas, sendo a Aprendizagem de Máquina (AM), do inglês, *Machine Learning* (ML) uma de suas principais subáreas exploradas nos Testes de Software. Ainda segundo Rouhiainen (ROUHIAINEN, 2018), Aprendizagem de Máquina é um aspecto da computação em que as máquinas têm a capacidade de aprender sem serem programadas para isso. Podemos dividir o AM nos três seguintes conceitos:

- Aprendizado supervisionado
- Aprendizado não supervisionado
- Aprendizado por reforço (menos corrente neste escopo)

Segundo Lorena e Carvalho(LORENA; DE CARVALHO, 2007), o aprendizado supervisionado consiste em um aprendizado onde são fornecidas referências através de um treinamento utilizando um conjunto de entradas e saídas para que o algoritmo adquira conhecimento com base nos dados utilizados. Já o aprendizado não supervisionado consiste em não utilizar valores de referências para que o algoritmo adquira conhecimento, ou seja, o algoritmo deverá aprender por conta própria sem passar por um treinamento.

Ainda segundo os mesmos autores, os dois tipos de aprendizado têm focos diferentes, o supervisionado visa gerar resultados com base nos dados treinados, fazendo com que o algoritmo crie um certo padrão e consiga produzir resultados nesse padrão usando novas entradas, enquanto o não supervisionado tem como objetivo agrupar dados semelhantes e criar tendências, já que não existe um parâmetro a ser seguido previamente.

Na hora de criar classificadores de Aprendizagem de Máquina, cuja função é treinar dados e fazer previsões a partir deles, é importante se atentar aos dados utilizados no conjunto de testes. Para construir modelos de alta qualidade é necessário dados de qualidade, evitando assim a ocorrência de vício ou sub-aprendizagem (do inglês *overfitting* e *underfitting*). O conceito de vício em AM (*overfitting*) se assemelha a uma situação em que o modelo de AM apresenta uma grande quantidade de acerto mas apenas com os dados de teste, com a inserção de novas entradas o classificador não consegue atingir resultados satisfatórios e desempenhar da forma esperada. Já a sub-aprendizagem (underfitting) seria o oposto dessa situação, nesse caso o classificador apresenta uma grande taxa de erro nos conjuntos de treino e de teste.

2.3 Intersecção das Áreas

Existe uma diversidade perceptível em relação à aplicação da Inteligência Artificial dentro do processo de testes de software, seja para aplicação dentro do processo ou simplesmente dentro do teste em si.

A IA pode potencializar os testes de diversas formas e tem um potencial promissor conforme destaca Hourani et al (HOURANI; HAMMAD; LAFI, 2019). Para aqueles autores as tendências da IA, baseadas em uma análise de pesquisas e estudo de previsões, incluem:

- A IA se fará presente em todo o processo de testes, desde o planejamento até a execução sem intervenções humanas;
- Os testes de software com IA se tornarão uma indústria com vários segmentos autônomos, onde engenheiros de testes serão substituídos pela IA e desempenharão uma nova função no monitoramento e resultados;

- A IA cobrirá os testes de diversas áreas (sites, banco de dados, mobile, jogos, etc);
- As novas ferramentas para testes com IA serão inovadoras, ágeis e inteligentes, e
- A análise preditiva de IA nos testes de software tornará os produtos mais robustos e confiáveis.

- Avaliar aspectos descritivos e quantitativos referentes a estes estudos, em especial a localidade, as instituições, meios de comunicação e tipos de estudos, com intuito de detectar predominância ou não de algum destes fatores.

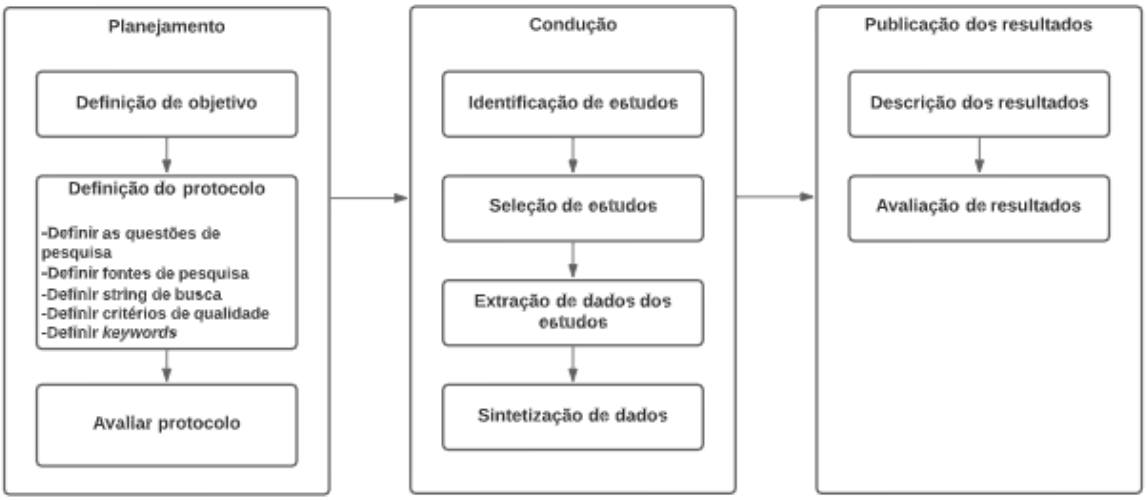

Figura 2 - Diagrama com as principais etapas do mapeamento e suas subtarefas segundo Falbo (FALBO, 2018).

Ainda segundo os autores, apesar da aplicação de IA ser predominante em estudos acadêmicos, ele já começa a crescer no âmbito industrial. O mercado impulsionado pela IA é muito amplo e em conjunto com a área de qualidade que vem em uma grande crescente, a tendência é que nos próximos anos os testes com IA se tornem essenciais dentro das grandes organizações.

3. Metodologia Empregada

Considerando a área de estudo anteriormente definida, a pesquisa deste mapeamento sistemático estabelece três objetivos principais:

- Reunir estudos primários e secundários sobre o uso de IA no processo de testes, tendo como critério principal a data mais recente de publicação;

- Analisar técnicas e as áreas de atuação em ambos os campos do saber: Testes de Software e Inteligência Artificial, que se destacam nos últimos 5 anos, e

Para realização desse trabalho de mapeamento sistemático, foi adotada a metodologia apresentada em Falbo (DE ALMEIDA FALBO, 2018) que por sua vez é fortemente baseada em Kitchenham e Charters (KITCHENHAM; CHARTERS, 2007). Essa metodologia consiste em três principais etapas: planejamento, condução e exposição de resultados, sintetizada na Figura 2.

Na etapa de Planejamento, são três as tarefas: definição de objetivos, já estabelecidos nos parágrafos anteriores; definição de protocolo, elementos contidos na seção 3.1; e avaliação do protocolo.

A etapa de Condução subdivide-se em: identificação de estudos através coleta por meio dos mecanismos de buscas; a seleção de estudos com a aplicação dos critérios estabelecidos; a extração dos dados considerando os objetivos; e a síntese dos dados tabulando-os. A última etapa denominada publicação dos resultados comporá a seção 5 deste estudo, com a elaboração de gráficos e tabelas e da discussão através das respostas às perguntas estabelecidas.

Como ferramentas de apoio para condução deste trabalho foram empregadas as seguintes

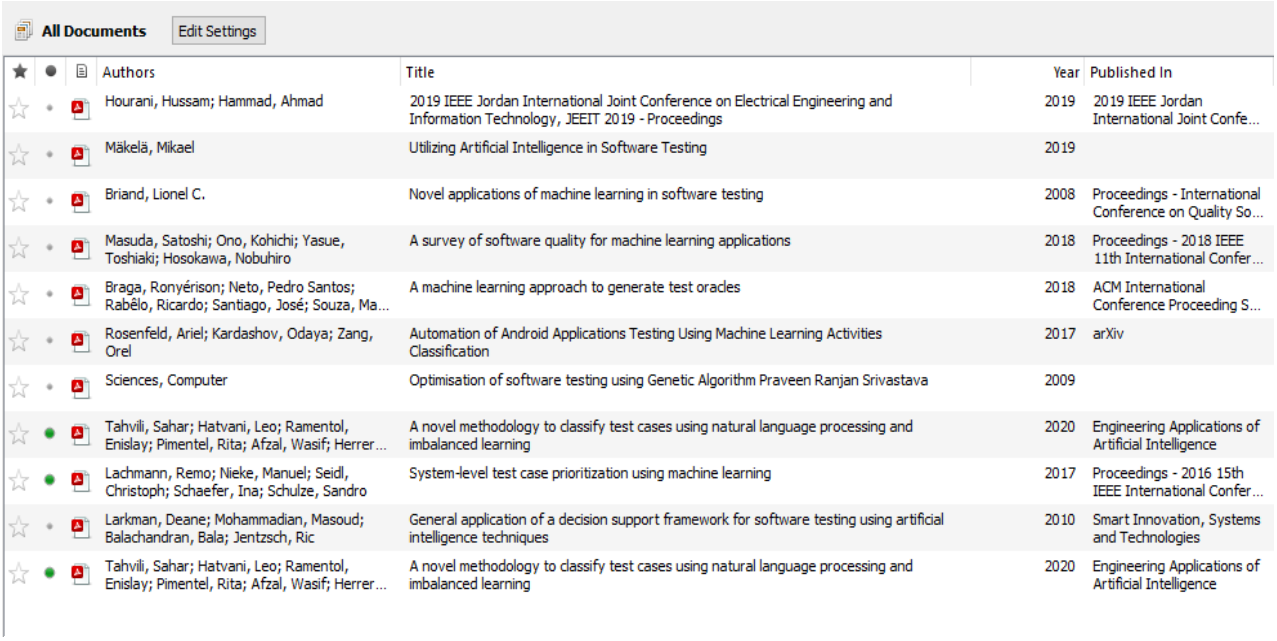

Figura 3 - Uso do Mendeley para listagem dos artigos iniciais.

ferramentas: Parsifal, software web para auxílio na execução de estudos sistematizados; Mendeley[2], Zotero[3] e Paperpile[4], para organização, marcação e anotações dos arquivos digitais correspondentes aos estudos selecionados; Documentos[5] do Google para escrita e tabulação dos dados; Exploratory[6], Colaboratory[7] e Microsoft Excel[8] para processamento estatístico e geração de elementos gráficos; e por fim, diferentes sistemas de buscas de referências diretas e semânticas de literatura acadêmica.

3.1 Protocolo

3.1.1 Questões de Pesquisa

Para elaboração das questões de pesquisa foi realizado uma análise inicial a partir de dez primeiros artigos sugeridos pelo buscador Google Scholar[9], ordenados a partir do critério de relevância estabelecido por ele, como resultado pela busca da string de texto "***'software test' and 'artificial intelligence'***". A razão por tal procedimento se deu em função de a ferramenta ser adotada como padrão *de fato* pela comunidade e por agregar milhões de artigos acadêmicos. Entretanto, ressalta-se que por não ser um procedimento replicável e por ter critérios de 'relevância' não explicitamente descritos, deve ser utilizada apenas para captar um momento instantâneo.

A Figura 3 mostra um segmento de imagem correspondente ao software Mendeley, com a listagem dos dez artigos utilizados. Inicialmente foram determinadas dez (10) questões iniciais, e em seguida foram reestruturadas e sintetizadas em cinco (5) Questões de Pesquisa (QP) finais, listadas como segue:

QP1: Quais atividades do processo de testes de software recebem maior foco ao utilizar intervenções de IA?

QP2: Quais tipos de testes de softwares são mais abordados pelas aplicações de IA?

QP3: Quais técnicas, algoritmos e áreas de IA são atualmente empregadas nas aplicações relacionadas aos testes de softwares?

QP4: Quais as possíveis consequências positivas (vantagens, benefícios, melhorias) e negativas (desvantagens, limitações) destacadas pelos autores

[2] Mendeley Desktop - https://www.mendeley.com/download-desktop/
[3] Zotero Research Assistance - https://www.zotero.org/
[4] Paperpile Reference Manager - https://paperpile.com/
[5] Google Documentos - https://docs.google.com/
[6] Exploratory Data Science - https://exploratory.io/
[7] Colaboratory Research - https://research.google.com/colaboratory/
[8] Microsoft Excel - https://office.live.com/start/Excel.aspx?ui=pt-BR
[9] Google Scholar - https://scholar.google.com/

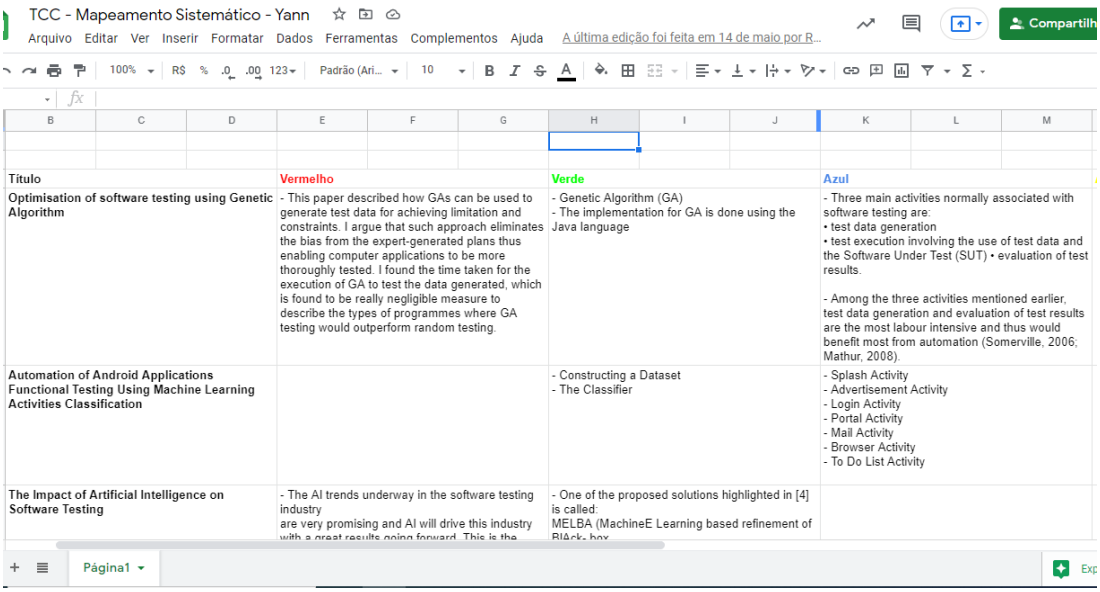

Figura 4 - Planilha utilizada para armazenar as informações importantes retiradas dos artigos pré-mapeamento.

na aplicação de IA no processo de testes de software?

QP5: Qual é a variedade metodológica das pesquisas relacionadas ao uso de IA no processo de testes de software neste momento? Existem grupos de destaque? Canais de comunicação predominantes?

Após definição das questões de pesquisa, como etapa de verificação, foi realizada uma prática de identificação de artefatos textuais nos próprios artigos na tentativa de identificar elementos que subsidiem as respostas às questões estabelecidas. A Figura 4 apresenta segmento de imagem com parte dos artefatos e corroborando a executabilidade para obtenção dos dados.

3.1.2 Fontes de Pesquisa

Os motores de busca escolhidos para obtenção dos artigos que compuseram o mapeamento foram escolhidos a partir dos agregadores identificados pelo Google Scholar durante a busca inicial, bem como por uma lista disponibilizada pelo software Parsifal, desde que acessíveis a partir do sistema de autenticação fornecido pelo site Periódicos da Capes (MEC - Governo Federal), do qual os autores têm acesso. A lista é composta por oito (8) sistemas: ACM Digital Library[10], EI Compendex[11], Google Scholar[12], IEEE Digital Library[13], ISI Web of Science[14], Science@Direct[15], Scopus[16] e Springer Link[17].

3.1.3 *String* de Busca

Diferentes *strings* de busca foram avaliadas e observou-se, empiricamente, que o conjunto que filtrou e permitiu a ocorrência dos artigos mais relacionados ao tema, limitando o total final a três mil resultados em um intervalo definido de cinco(5) anos, foi a seguinte:

I. ("Software Testing") **AND** ("Artificial Intelligence" **OR** "Machine Learning")

II. ("Software Testing" **AND** "Artificial Intelligence") **OR** ("Software Testing" **AND** "Machine Learning")

Dependendo do motor de busca a *string* precisou sofrer reestruturação da apresentação I, para a apresentação II para se adequar a cada um deles. Alguns sistemas utilizam uma notação funcional, enquanto outros o uso da notação direta é suficiente.

[10] ACM Digital Library - http://portal.acm.org
[11] EI Compendex - http://www.engineeringvillage.com
[12] Google Scholar - https://scholar.google.com.br
[13] IEEE Digital Library - http://ieeexplore.ieee.org
[14] ISI Web of Science - http://www.isiknowledge.com
[15] Science@Direct - http://www.sciencedirect.com
[16] Scopus - http://www.scopus.com
[17] Springer Link - http://link.springer.com

3.1.4 Definição dos Critérios de Qualidade (Inclusão e Exclusão)

Dos critérios utilizados para inclusão de artigos/estudos na lista a ser avaliada, estão:

1. Deverá dissertar sobre inteligência artificial aplicado aos testes de software
2. Ter sido revisado por pares (periódicos ou congressos)
3. Conter artefatos relacionados à resposta de pelo menos uma das questões de pesquisa
4. Pertencer ao intervalo de cinco (5) anos mais recentes e dentre os cem (cem) mais recentes considerando o motor de busca originário
5. Estar entre os cinquenta (50) trabalhos mais recentes considerando todos os motores de busca

O quarto critério de inclusão garante que no máximo cem (100) estudos de cada motor de busca participem da seleção, por critério de data, e ao mesmo tempo garante que cada motor de busca contribuirá igualitariamente. O quinto critério impõe um limite máximo em função do tempo e esforço de trabalho factível durante a execução do TCC. Dos critérios utilizados para exclusão de artigos/estudos na lista a ser avaliada estão:

1. Não conter um resumo no corpo inicial do texto
2. Ser um estudo do tipo Resumo apenas
3. Ser uma *pre-print* sem revisão por pares

3.2 Condução do Mapeamento

Usando as *strings* de busca definidas foi possível obter um total de 637 estudos, dos quais foram escolhidos 50, conforme o procedimento replicável seguinte. Para cada motor de busca foi escolhido a busca avançada com a inserção da *string* correspondente, limitando o escopo temporal entre o anos de 2015 e o primeiro semestre de 2021, ordenando o resultado por data (mais recente) sem utilizar busca semântica, considerando todos os campos disponíveis (título, resumo, keywords, texto). A Tabela 1 mostra os quantitativos para cada motor.

Tabela 1 - Resultados da obtenção de estudos por motor de busca

Motor	**Total**	**Selecionados**
ACM Digital Library	1170	100
Google Scholar	40	40
IEEE Digital Library	396	100
ISI Web of Science	174	100
Science@Direct	98	98
Scopus	1536	100
SpringerLink	2161	99

Após a identificação dos estudos a partir dos motores de busca, utilizou-se do Parsifal como ferramenta de apoio à aplicação dos critérios de qualidade (inclusão e exclusão) dos estudos. Para isso, todos os estudos foram ordenados pelo critério de data (mais recente para o mais antigo).

Como resultado do processo de seleção foram exportados a listagem de cinquenta (50) estudos, contendo título, autores, ano e DOI (identificador único) quando presente. Os estudos foram importados na ferramenta Paperpile, onde foram filtrados com relação aos critérios de exclusão. Dos cinquenta estudos, cinco (5) estudos foram excluídos, um (1) foi excluído por se tratar de tese, outro (1) por se tratar de resumo apenas e um por estar fora do escopo (fora do tema). Três (3) estudos eram estudos tipo *pre-print*, ou seja, sem revisão por pares, e sete (7) não estavam acessíveis para serem avaliados, nem através dos sistemas de autenticação nem mesmo através de mecanismos de buscas gerais e redes sociais. Ao final, trinta e sete (37) estudos formaram a base desta pesquisa.

A Tabela 2, apresenta uma versão reduzida contendo as principais informações de identificação dos estudos primários. A Tabela 3 é a equivalente para os estudos secundários. Vale destacar que o termo "EP" demarca o estudo como um estudo primário (totalizando 26) e "ES" como um estudo secundário (totalizando 11).

Tabela 2 - Lista de estudos selecionados.

Tipo	Título	DOI
EP 01	A Comparison Experiment On Software Quality Testing With Machine Learning Using Different Algorithms	-
EP 02	A Conceptual Framework for Software Fault Prediction Using Neural Networks	10.1007/978-3-030-39237-6_12
EP 03	A Novel approach of GUI Mapping with image based widget detection and classification	10.1109/ICIEM51511.2021.9445281
EP 04	A novel methodology to classify test cases using natural language processing and imbalanced learning	10.1016/J.ENGAPPAI.2020.103878
EP 05	An ensemble-based predictive mutation testing approach that considers impact of unreached mutants	10.1002/STVR.1784
EP 06	Automatically predicting bug severity early in the development process	10.1145/3377816.3381738
EP 07	Boosted exploratory test architecture: Coaching test engineers with word similarity	10.1109/ICSTW52544.2021.00038
EP 08	Boosting Exploratory Testing of Industrial Automation Systems with AI	10.1109/ICST49551.2021.00048
EP 09	Code Coverage Similarity Measurement Using Machine Learning for Test Cases Minimization	10.1109/GCCE50665.2020.9291990
EP 10	Defect Prediction With Semantics and Context Features of Codes Based on Graph Representation Learning	10.1109/TR.2020.3040191
EP 11	Just-in-Time Defect Prediction for Android Apps via Imbalanced Deep Learning Model	10.1145/3412841.3442019
EP 12	Large Scale Evaluation of Natural Language Processing Based Test-to-Code Traceability Approaches	10.1109/ACCESS.2021.3083923
EP 13	Learning to Generate Fault-revealing Test Cases in Metamorphic Testing	10.1016/j.jss.2020.110574
EP 14	Learning to Predict Software Testability	10.1109/CSICC52343.2021.9420548
EP 15	Method of Robotic Process Automation in Software Testing Using Artificial Intelligence	10.1109/ACIT49673.2020.9208806
EP 16	MuDelta: Delta Oriented Mutation Testing at Commit Time	10.1109/ICSE43902.2021.00086
EP 17	On Learning Meaningful Assert Statements for Unit Test Cases	10.1145/3377811.3380429
EP 18	Optimization of association rules using hybrid data mining technique	10.1007/S11334-021-00387-6
EP 19	Prioritized Test Generation Guided by Software Fault Prediction	10.1109/ICSTW52544.2021.00045
EP 20	Software fault prediction based on the dynamic selection of learning technique: findings from the eclipse project study	10.1007/S10489-021-02346-X
EP 21	Software Visualization and Deep Transfer Learning for Effective Software Defect Prediction	10.1145/3377811.3380389
EP 22	Test case prioritization to examine software for fault detection using PCA extraction and K-means clustering with ranking	10.1007/S00500-020-05517-Z
EP 23	The Comparison of Feature Selection Methods in Software Defect Prediction	10.1109/ICICOS51170.2020.9299022
EP 24	Tool Support for Refactoring Manual Tests	10.1109/ICST46399.2020.00041
EP 25	What Are We Really Testing in Mutation Testing for Machine Learning A Critical Reflection	10.1109/ICSE-NIER52604.2021.00022
EP 26	What is AI Software Testing? and Why	10.1109/SOSE.2019.00015

Tabela 3 - Lista de estudos secundários.

Tipo	Título	DOI
ES 01	A detailed survey on machine intelligence based frameworks for software defect prediction	10.1109/ICCCIS51004.2021.9397136
ES 02	A Survey of different machine learning models for software defect testing	-
ES 03	A Systematic Literature Mapping of Artificial Intelligence Planning in Software Testing	10.5220/0009829501520159
ES 04	AI for Testing Today and Tomorrow: Industry Perspectives	10.1109/AITEST.2019.000-3
ES 05	AI-based test automation: A grey literature analysis	10.1109/ICSTW52544.2021.00051
ES 06	Artificial Intelligence Applied to Software Testing: A Literature Review	10.23919/CISTI49556.2020.9141124
ES 07	Automating and Optimizing Software Testing using Artificial Intelligence Techniques	10.14569/IJACSA.2021.0120571
ES 08	Integration of properties of virtual reality, artificial neural networks, and artificial intelligence in the automation of software tests: A review	10.1002/SMR.2159
ES 09	Machine Learning Applied to Software Testing: A Systematic Mapping Study	10.1109/TR.2019.2892517
ES 10	Machine Learning for Automation Software Testing Challenges, Use Cases Advantages & Disadvantages	-
ES 11	Role of Machine Learning & Artificial Intelligence Techniques in Software Testing	10.17762/TURCOMAT.V12I6.5800

Sobre a etapa de extração de dados (artefatos) e sintetização deste, após a escolha dos artigos finais, recolhemos as informações principais e passamos essas informações para uma planilha, listando informações como autores, journals em que foram publicados, ano de publicação, keywords e tópicos relacionados. Em seguida, utilizamos a ferramenta Paperpile para fazer marcações nos artigos e reunir as informações necessárias para responder às questões de pesquisa, utilizando da mesma técnica descrita anteriormente onde os artefatos correspondentes a cada pergunta é representada por uma cor.

4. Resultados e Discussão

O relato que se segue é uma descrição dos artefatos encontrados em função de cada uma das questões de pesquisas propostas, especialmente considerando os estudos primários. Ao final da seção pontua-se brevemente os estudos secundários.

4.1 Questão de Pesquisa 1

Quais atividades do processo de testes de software recebem maior foco ao utilizar intervenções de IA?

Considerando a estrutura sugerida por(SOMMERVILLE, 2019), as atividades do processo de testes de software que mais ocorrem nos estudos são os referentes ao Projeto de casos de teste e à preparação de dados de casos de teste. Equivalentemente, considerando as etapas na estrutura de(OLSEN et al., 2018), as atividades com maior incidência de experimentos são as de planejamento, análise e modelagem.

Os dados dos estudos apontam que as fases iniciais do processo de testes de software lideram na implementação de IA. Sendo mais comum na etapa de projeto e planejamento, especialmente que aplicam estratégias de classificação para escolha de testes [**EP 03, EP 04, EP 09, EP 14, EP15, EP 16, EP 20, EP 23**] e de cobertura de códigos [**EP 09, EP 14, EP 19**]. Em relação aos que tratam de análise e modelagem destacam-se os que se relacionam a geração de testes automaticamente [**EP 07, EP 17, EP 19, EP 24**]. Todas as demais etapas apresentam experimentos, porém em frequência reduzida, contendo apenas um ou nenhum artigo.

Tabela 4 – Atividades de testes com mais uso de IA

Atividade do Processo de Testes	Ocorrência
Planejamento	11
Análise e Modelagem	4

4.2 Questão de Pesquisa 2

Quais tipos de testes de softwares são mais abordados pelas aplicações de IA?

Com relação aos tipos, é evidente a predileção pelos testes de **detecção de falhas** [**EP 02, EP 10, EP 13, EP 18-20, EP 23**]. Geralmente estes são experimentos com testes de caixa branca, ou seja, aqueles que possuem acesso ao código, portanto os testes de predição de defeitos são realizados nessa etapa.

Em seguida vem os **testes de mutação** [**EP 05, EP 16, EP 25**], esses testes tem como objetivo forçar erros no código alterando suas estruturas para que o teste quebre ao ser executado, cada alteração gera uma nova mutação e a cada erro no código uma mutação é excluída, para o teste ser validado é necessário que todas as mutações sejam avaliadas.

Outros tipos de testes com mais de uma ocorrência, contendo até três(3) são: automação de testes, teste de cobertura, testes unitários, teste de interface, teste de integração, teste de disponibilidade.

4.3 Questão de Pesquisa 3

Quais técnicas, algoritmos e áreas de IA são atualmente empregadas nas aplicações relacionadas aos testes de softwares?

Para responder a esta questão foram definidas duas atividades de coleta de artefatos. Uma relacionada aos algoritmos mais frequentemente utilizados e outra com relação ao tipo de objetivo/aplicação de IA utilizada. Por existir uma variedade grandiosa de algoritmos nesta área, optou-se por destacar apenas os mais frequentes e aparições não esperadas.

Os algoritmos mais utilizados são os clássicos da área de aprendizagem de máquina: ***Decision Tree* e *ensembles*** [**EP 01, EP 08-09, EP 14, EP 16, EP 19-20**], **Redes Neurais Artificiais Clássicas (*Multilayer Perceptrons*)** [**EP 02, EP 06, EP 08, EP 14, EP 16, EP 20**], ***Random Forest*** [**EP 01, EP 05, EP 08, EP 14, EP 16], SVM [EP 04, EP 08-09, EP 20, EP 23**], ***K-Means*** [**EP 08, EP 18, EP 20, EP 22, EP 24**], são os mais utilizados. Destacam-se ainda algoritmos utilizados junto ao **processamento de linguagem (códigos)** [**EP 04, EP 06-07, EP 10, EP 12, EP 17**], e alguns não habituais neste segmento e com apenas uma aparição, são eles: **Lógica Fuzzy** [**EP 04**], **Algoritmos Genéticos** [**EP 19**] e **Aprendizagem por Reforço** [**EP 13**]. Como era de se esperar, algoritmos mais contemporâneos e robustos apareceram, porém com uma frequência menor que o esperado, são eles: ***Deep Learning*** [**EP 11, EP 21**], **Redes Convolucionais** [**EP 01, EP 21**], e **Redes Recorrentes** [**EP 17]**.

Sobre os principais tópicos, não há uma classificação mutuamente exclusiva, ao contrário, pode ocorrer complementaridade. A maioria dos trabalhos empregam o uso das técnicas clássicas de ***Aprendizagem de Máquina*** (**08 estudos**), com algoritmos já bem sedimentados e estáveis. Logo em seguida, ***Deep Learning*** é a escolha com mais ocorrências contendo **cinco(5) estudos**. Sobre os objetivos de aplicação priorizam-se os clássicos como é de esperar **Classificação** [**EP 03-04, EP 09, EP 14-16, EP 20, EP 23**], **Predição** [**EP 01-02, EP 05-06, EP 18-19, EP 21**] **e Clusterização**[**EP 08, EP 18, EP 22, EP 24**]. Ocorrem como inovação as aplicações de **Detecção de Objetos (Visão Computacional)** [**EP 03, EP 15**] e **de Criação de Conteúdo (NLP)** [**EP 07, EP 17, EP 19, EP 24**].

A Tabela 5 sumariza a frequência de ocorrência nos estudos dos algoritmos de IA e dos tópicos presentes.

Tabela 5 - Frequência de ocorrências de algoritmos e tópicos de estudo de IA

Algoritmo de IA	**Ocorrência**
decision tree	7
redes neurais	6
random forest, svm, k-means	5
logistic regression, knn, doc2vec, boosting	3
gradient boosting, cnn, word2vec, dnn, adaboost	2
bagging classifier, bernoulli-nb, imbalanced algorithms, fuzzy, gaussian process regression, dbscan, adaboostedtree, gnn, lda, idl, lsi, ensemble, reinforcement learning, rnn, lstm, a priori, genetic algorithm, naïve bayes, pca, ranking, relieff, mean-shift	1

Tópico de IA	Ocorrência
machine learning, classificação	8
predição	7
deep learning, nlp	5
clusterização, geração de conteúdo	4
detecção de objetos, imbalanced learning	2
visão computacional, redução de dimensionalidade, regressão, active learning, graph representation learning, reinforcement learning, ordenação, feature selection	1

4.4 Questão de Pesquisa 4

Quais as possíveis consequências positivas (vantagens, benefícios, melhorias) e negativas (desvantagens, limitações) destacadas pelos autores na aplicação de IA no processo de testes de software?

O benefício/vantagem mais citado é a otimização/**ganho de tempo** que as técnicas de IA podem proporcionar. Entretanto, outros também foram citados mais de uma vez, incluindo:

- Robustez maior devido a "**prevenção de falhas**";
- **Redução do trabalho** repetitivo humano;
- Adaptabilidade dos algoritmos de IA em relação às mudanças de código, **evitando o retrabalho**;
- Agilidade na **automação de testes** e tarefas relacionadas, e
- Melhoria na **detecção de falhas** e de requisitos mal projetados.

O uso da IA é bastante diversificado e garante inúmeras vantagens no processo de testes de software, entretanto também existem algumas desvantagens e algumas puderam ser extraídas a partir dos estudos:

- Dificuldade em resolver padrões de **problemas não previamente modelados** que surgem no desenvolvimento de softwares, neste caso, é preciso de um pouco mais de tempo para a IA ter um melhor amadurecimento dentro da fase de testes, e
- Ainda será **necessário intervenção humana** mesmo para atividades de nível de complexidade médio, e
- É uma área que ainda está em exploração, portanto ainda **não existem soluções** bastante efetivas e consolidadas para aplicações no dia a dia em um ambiente empresarial.

4.5 Questão de Pesquisa 5

Qual é a variedade metodológica das pesquisas relacionadas ao uso de IA no processo de testes de software neste momento? Existem grupos de destaque? Canais de comunicação predominantes?

Uma primeira informação acerca dos estudos obtidos diz respeito à distribuição temporal e os tipos de estudos. Pelo próprio protocolo priorizou-se os estudos publicados nos últimos cinco(5) anos. O gráfico 1, apresenta um gráfico da distribuição dos estudos em relação aos anos e aos tipos de artigos. Percebe-se que há uma grande produção em andamento tendo em vista que há uma quantidade maior de estudos de 2021 (21 no total), e considerando que a coleta foi realizada no primeiro semestre de 2021, e comparado aos estudos de 2019 e 2020 (16 no total). Informação relevante também de que a maioria dos estudos de 2021 são de revistas/jornais, geralmente esses conteúdos são mais abrangentes.

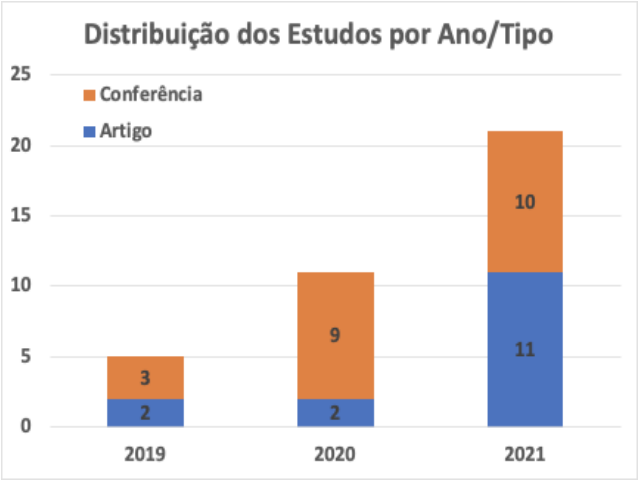

Gráfico 1 - Distribuição de Estudos por Ano e Tipo

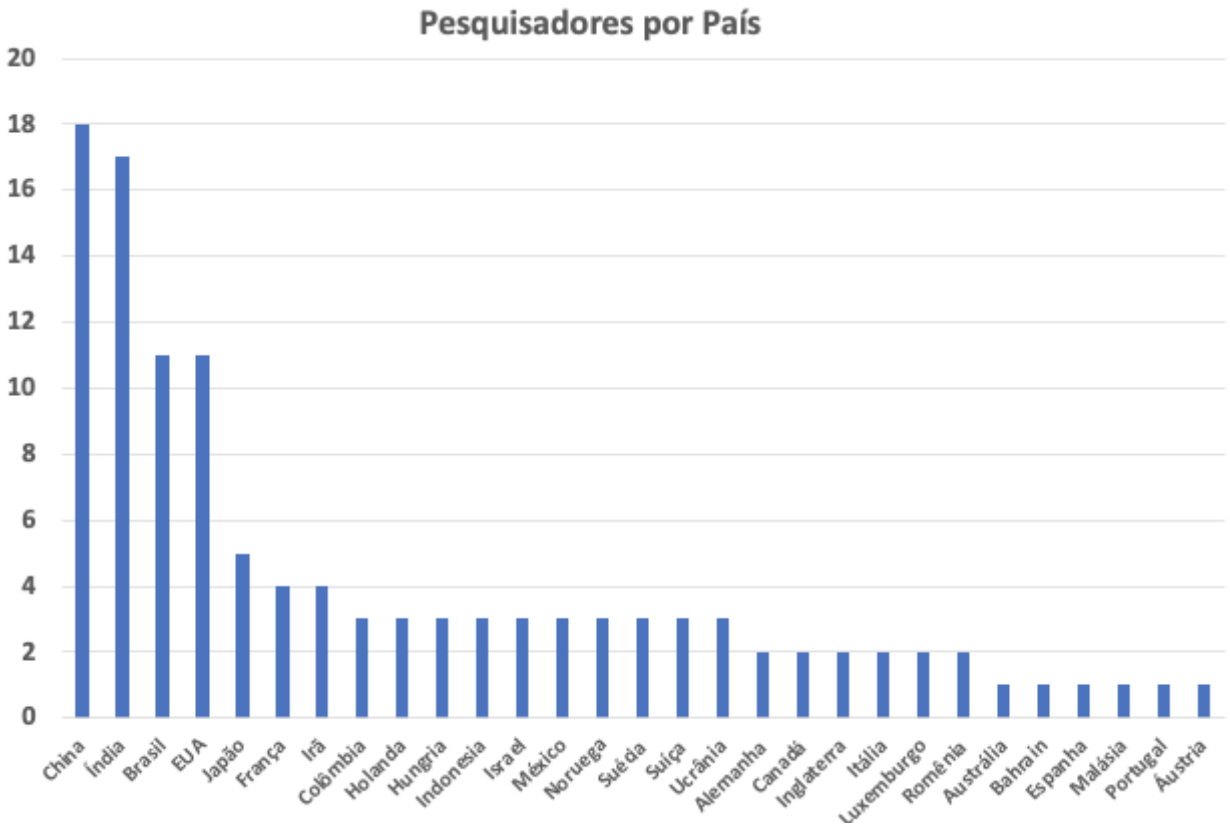

Gráfico 2 - Quantidade de Pesquisadores por País.

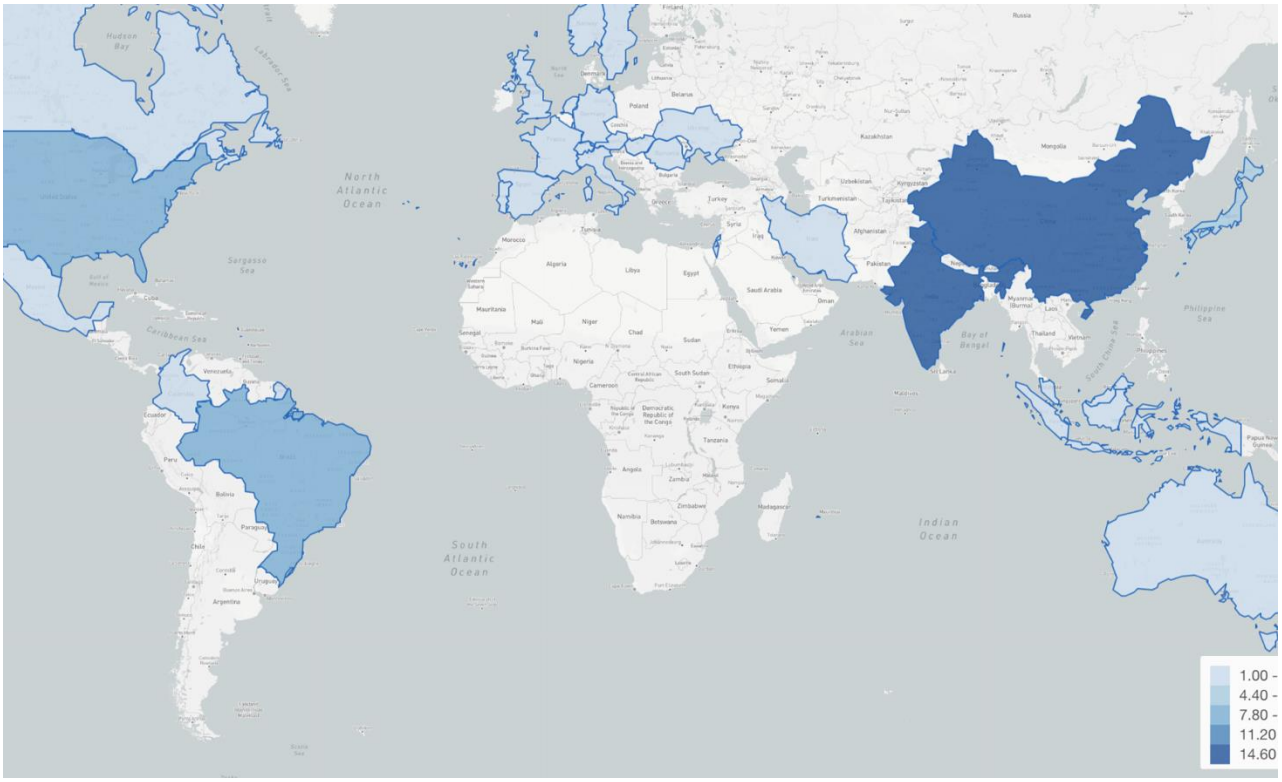

Figura 5 - Distribuição das contribuições por continentes.

Ou seja, não há uma prioridade na escolha dos canais em relação ao tipo de conteúdo, estando ambas as comunicações (revistas, conferências) em equilíbrio, com exceção de 2020.

Do total de estudos, há apenas um estudo por autor, ou seja, nenhum autor/pesquisador aparece mais de uma vez na lista total. Devido a este fato, optou-se por considerar as contribuições individuais ao se comparar dados 'demográficos' relativos a países e instituições de filiação. No gráfico 2, é possível ver a quantidade de pesquisadores autores por país. E na Figura 5, a mesma informação considerando a distribuição através dos continentes.

Da figura 5, percebe-se que os países Asiáticos em especial China e Índia contribuem em maior quantitativo de pesquisadores, que toda a Europa está bastante presente, que nas Américas o Brasil e EUA predominam, que há participação relevante da Oceania, porém uma ausência marcante e significativa de países Africanos.

Com relação à variedade de tipos de estudos, têm-se onze (11) Estudos Secundários e vinte e seis (26) estudos primários. Sobre os últimos, três (3) são estudos de proposições teóricas e discursivas, vinte e quatro (24) são experimentos, e destes, cinco (5) também apresentam ferramentas/frameworks como parte da contribuição.

Com relação às instituições, a chinesa Zhejiang University of Technology é a que apresenta o maior número de pesquisadores contribuintes (5), seguido da brasileira Universidade Federal do Paraná (4). Sobre os tipos de afiliações, como se imaginava, a maioria é membro de instituições acadêmicas (universidades, faculdades, centros de pesquisas atrelados a universidades ou governo) totalizando cento e seis (106), aparecem também pesquisadores de origem empresarial/industrial totalizando doze (12), e até mesmo um pesquisador independente sem filiação industrial específica nem acadêmica. Considerando a nacionalidade, o maior quantitativo de acadêmicos presentes é oriundo da China (18), Índia (15) e Brasil (11), já o total de pesquisadores da indústria são em sua maioria dos EUA (6) e Suíça (3).

4.6 Estudos Secundários

Dentre os trinta e sete estudos (37), onze são estudos secundários. Cada um destes contribui especialmente com a compreensão da área. Importante frisar que este estudo desenvolvido se difere dos demais porque intenta retratar o momento atual, o critério temporalidade é o que define este estudo e a interpretação dos dados, bem como a extração destes sustenta-se em interpretação quantitativa. Entretanto, optou-se por descrever brevemente cada um dos estudos secundários identificados:

- **ES 01 -** Esse estudo tem como objetivo identificar e compreender as estratégias existentes para prever defeitos usando inteligência artificial.
- **ES 02 -** Discute diferentes tipos de sistemas de previsão de defeitos junto com suas limitações.
- **ES 03 -** Esse mapeamento sistemático da literatura traz uma discussão sobre de 16 artigos contendo propostas de métodos, técnicas e ferramentas utilizadas para se fazer uso de IA dentro do processo de testes de software.
- **ES 04 -** Esse estudo reúne informações sobre visões, ideias, pensamentos e perspectivas de diferentes autores a respeito da área de inteligência artificial em testes de software.
- **ES 05 -** Esse estudo traz uma literatura cinzenta sobre o uso de IA para automação de testes. Segundo os autores, o estudo reuniu cerca de 136 documentos visando identificar problemas que a IA possa resolver dentro da automação de testes, o artigo expõe 6 ferramentas utilizadas para auxiliar na automação de testes e faz uma crítica sobre o estado atual e o que se espera do uso de IA em testes de software.
- **ES 06 -** Essa revisão literária traz de forma resumida o estado atual do uso de IA nos testes de software durante os últimos 3 anos, reunindo materiais das bases Scopus Elsevier, web of Science e Google Scholar. O estudo tem como objetivo descobrir as principais abordagens de IA e em quais tipos de testes elas são utilizadas.
- **ES 07 -** O estudo mostra de forma resumida o estado atual da área de testes e descreve várias técnicas de testes e IA que podem ser usados juntos
- **ES 08 -** Esse estudo discute um possível cenário onde testes poderão ser 100% automatizados, e para esse cenário ser possível, os autores analisam o uso de realidade virtual, redes neurais e inteligência artificial.
- **ES 09 -** O mapeamento sistemático reuniu 48 artigos com a intenção de descobrir como a Machine Learning tem atuado para otimizar os testes de software, trazendo uma visão geral sobre esses dois temas. Os resultados trazem os algoritmos mais usados para realizar testes de software e discussões sobre as pesquisas atuais nesses dois ramos, os autores afirmam que existe

uma necessidade de melhores estudos empíricos sobre esse determinado tema.

- **ES 10 -** No estudo discute-se as vantagens, desvantagens e desafios da automação de testes usando machine learning.
- **ES 11 -** Neste objetivou-se explorar o uso de técnicas de machine learning em testes de software e explicar de forma detalhada a implementação destas técnicas.

5. Considerações Finais

O futuro para a evolução dos testes de software com a implementação de Inteligência Artificial é bastante promissor, a diversidade de estudos a respeito é muito grande. Observou-se, neste momento, a predominância de estudos sobre a prevenção de defeitos, provavelmente pela facilidade em utilizar as abordagens da IA. Apesar da predominância, é possível observar espaço e avanços em diversos tipos e etapas dos processos de teste de softwares. Com os dados obtidos neste trabalho, podemos afirmar que todos os principais tipos de teste que são utilizados na indústria já têm implementações práticas com intervenção de alguma técnica de IA. As pesquisas se demonstraram bastante variadas, principalmente no uso de todas as técnicas clássicas de IA e também das mais recentes, nomeadamente Deep Learning, Redes Convolucionais e Processamento Natural de Linguagem.

Este último, Processamento de Linguagem Natural (PLN), em especial, destaca-se pela capacidade de atuar na modelagem de testes em nível de criação de códigos, indo além da detecção ou classificação dos destes. Ainda foi possível verificar a aplicação da técnica para reconhecer erros em cenários de teste e refatorá-los, mesmo que com algumas intervenções humanas pontuais.

Em termos de ferramentas, ainda não existe uma ferramenta específica para se fazer testes abrangentes e genéricos com uso de IA, existem algumas técnicas e ferramentas que tornam possíveis o uso da IA em domínios pontuais e específicos, até dependentes de linguagens, mas vejo todas com potencial para se firmarem e serem usadas no âmbito empresarial.

Alguns autores acreditam que em um futuro próximo chegaremos a ter 100% de automação nos testes de software. A IA seria o principal fator para que esse cenário se concretize, as pesquisas estão frequentes e em crescimento quantitativo.

Acredita-se que este estudo pode ser facilmente replicável, tornando viável a expansão do mesmo para uma quantidade maior de estudos ou mesmo um ajuste temporal no escopo de coleta e tratamento dos dados.

Referências

DE ALMEIDA FALBO, R. Mapeamento sistemático. **Retrieved October**, v. 7, 2018.

HOURANI, H.; HAMMAD, A.; LAFI, M. **The impact of artificial intelligence on software testing**. 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). **Anais**...IEEE, 2019.

JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. **Science**, v. 349, n. 6245, p. 255–260, 17 jul. 2015.

KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. 2007.

LORENA, A. C.; DE CARVALHO, A. C. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43–67, 2007.

OLSEN, K. et al. Certified tester foundation level syllabus. **Proc. Int. Softw. Testing Qualifications Board (ISTQB)**, p. 1–96, 2018.

RAMOS, I. A. **Predição de defeitos just-in-time em software utilizando inteligência artificial**. Sobral: Universidade Federal do Ceará, 2020.

ROUHIAINEN, L. **Artificial Intelligence: 101 things you must know today about our future**. [s.l.] Lasse Rouhiainen, 2018.

SOMMERVILLE, I. **Engenharia de Software**. 10. ed. São Paulo: Pearson, 2019.